SAND2023-05405C

Sandia National Laboratories

Exceptional service in the national interest

The Promise of Neuromorphic Computing

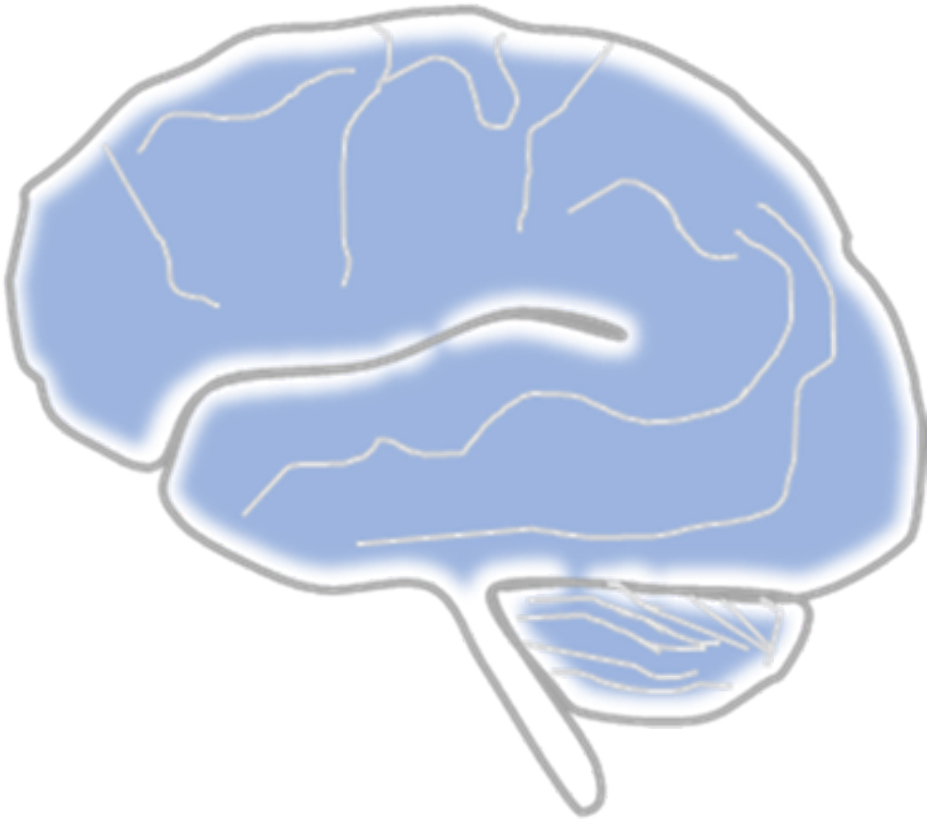
Rob Hoekstra

June 26, 2023

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

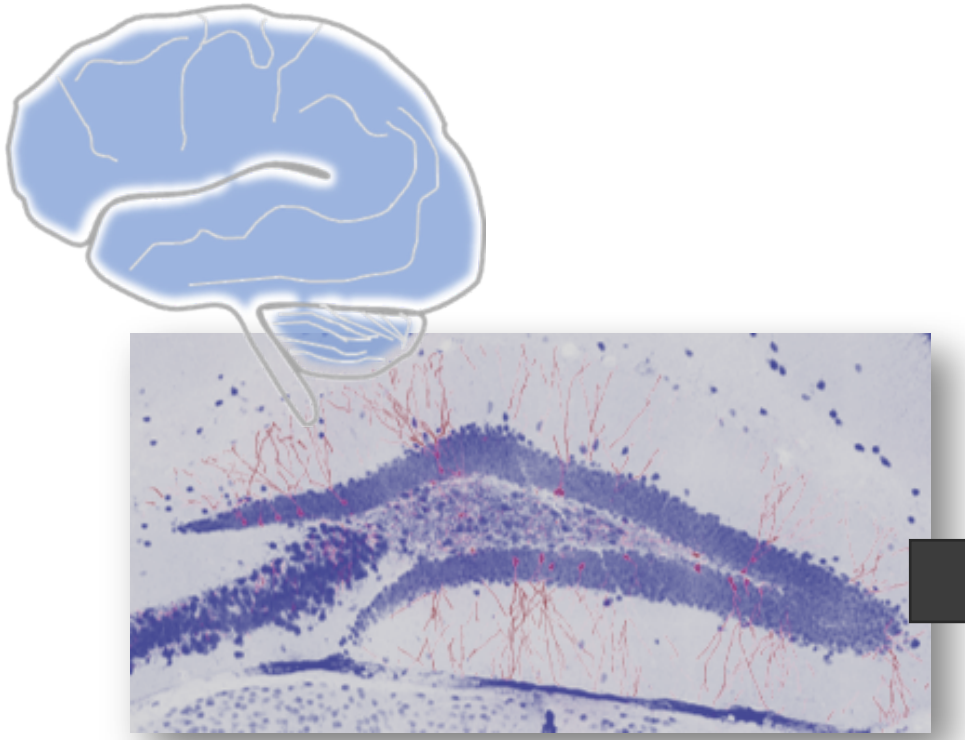
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

So why the brain?



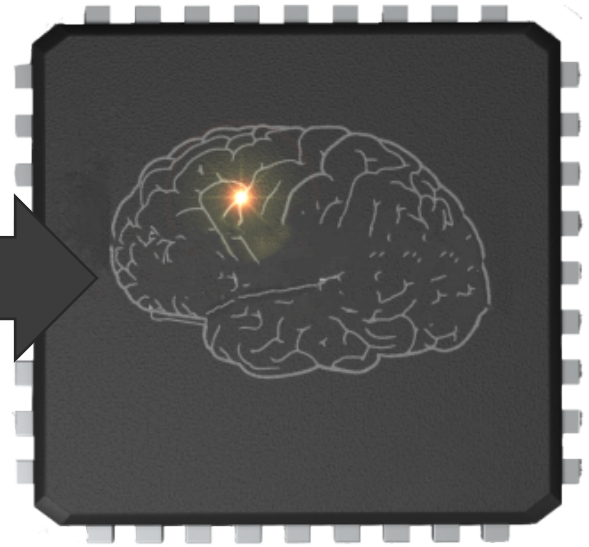
- Energy efficient
- Operationally fast considering slow components
- Data efficient
- Diverse applications
- Robustness

What is the brain good at?



Realized Features of Brain Inspiration in Neuromorphic Hardware

- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity

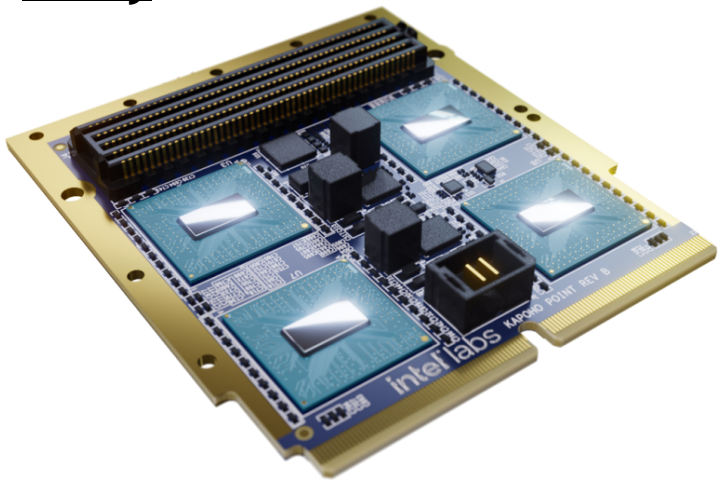


Where are we today in neuromorphic computing?

Realized Features of Brain Inspiration in Neuromorphic Hardware

- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity

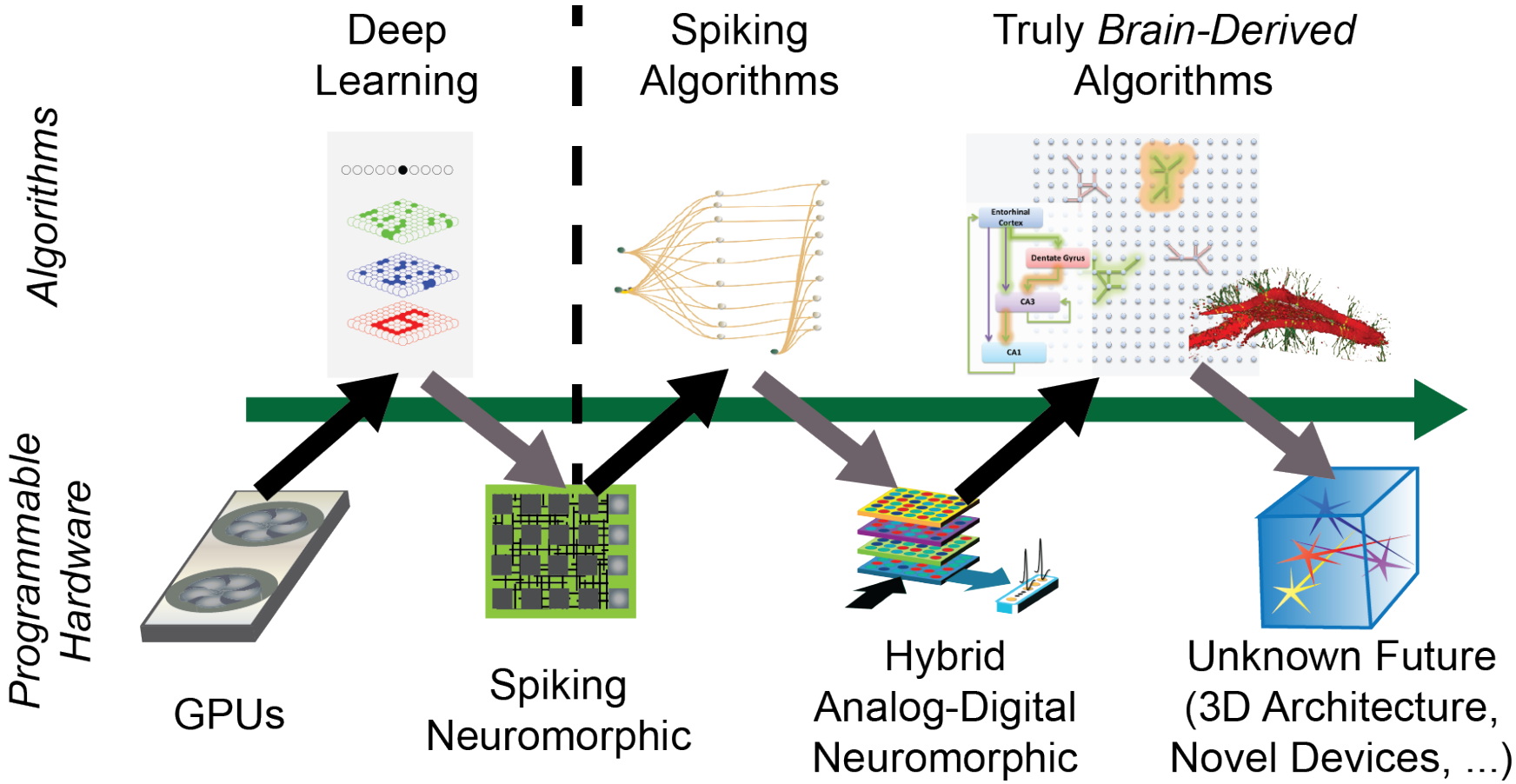
Today



Intel Loihi 2
Millions of CMOS neurons
Billions of CMOS synapses
~ 1 Watt power

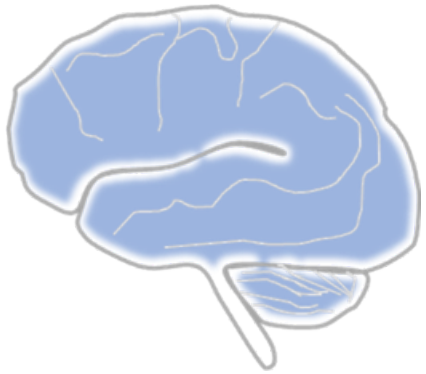
One example (of many…)

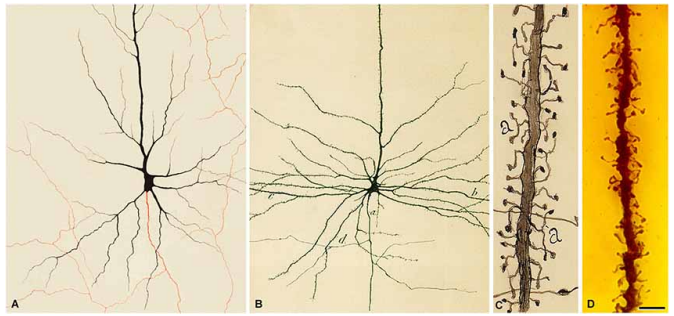
Towards novel future architectures



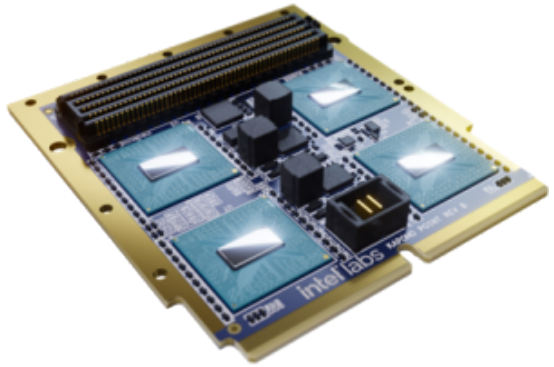
Aimone JB, Advanced Intelligent Systems, 2023

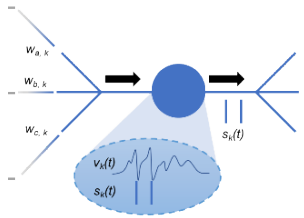
Fundamental science question: what is the scale of neuromorphic computing needed?



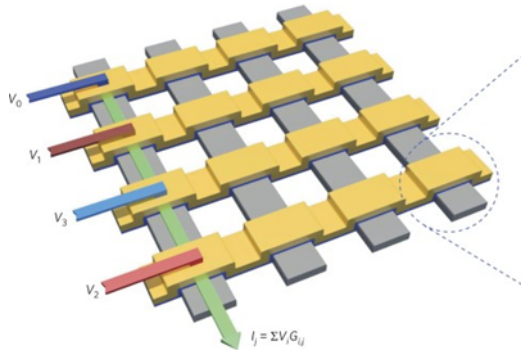


>10^{11} neurons
>10^{15} synapses
High complexity
Highly efficient
Slow





>10^{5} neurons
>10^{8} synapses
Low complexity
Moderately efficient
Fast



I=ΣV/R

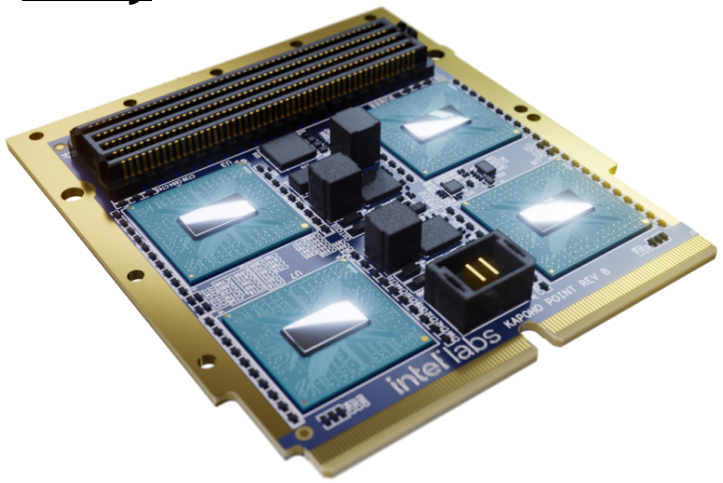
~10^{2} neurons
~10^{4} synapses
Very low complexity
Highly efficient
Fast?

Where are we today in neuromorphic computing? And where might we be in the future?

Realized Features of Brain Inspiration in Neuromorphic Hardware

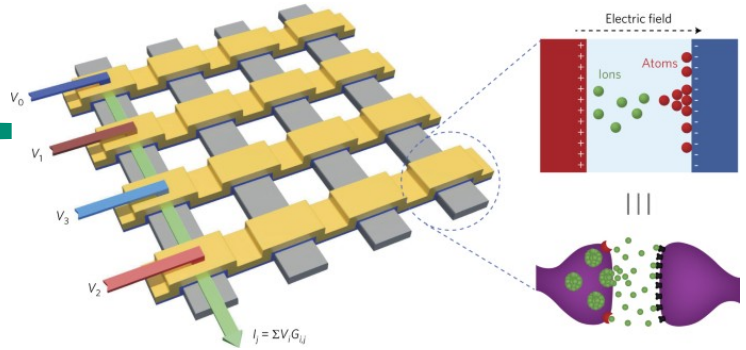
- Event-driven communication
- Graph based connectivity
- Processing in Memory
- In situ learning
- Analog computation
- Post-Moore's Law Devices
- Ubiquitous stochasticity

Today



Intel Loihi 2
Millions of CMOS neurons
Billions of CMOS synapses
~ 1 Watt power

Tomorrow

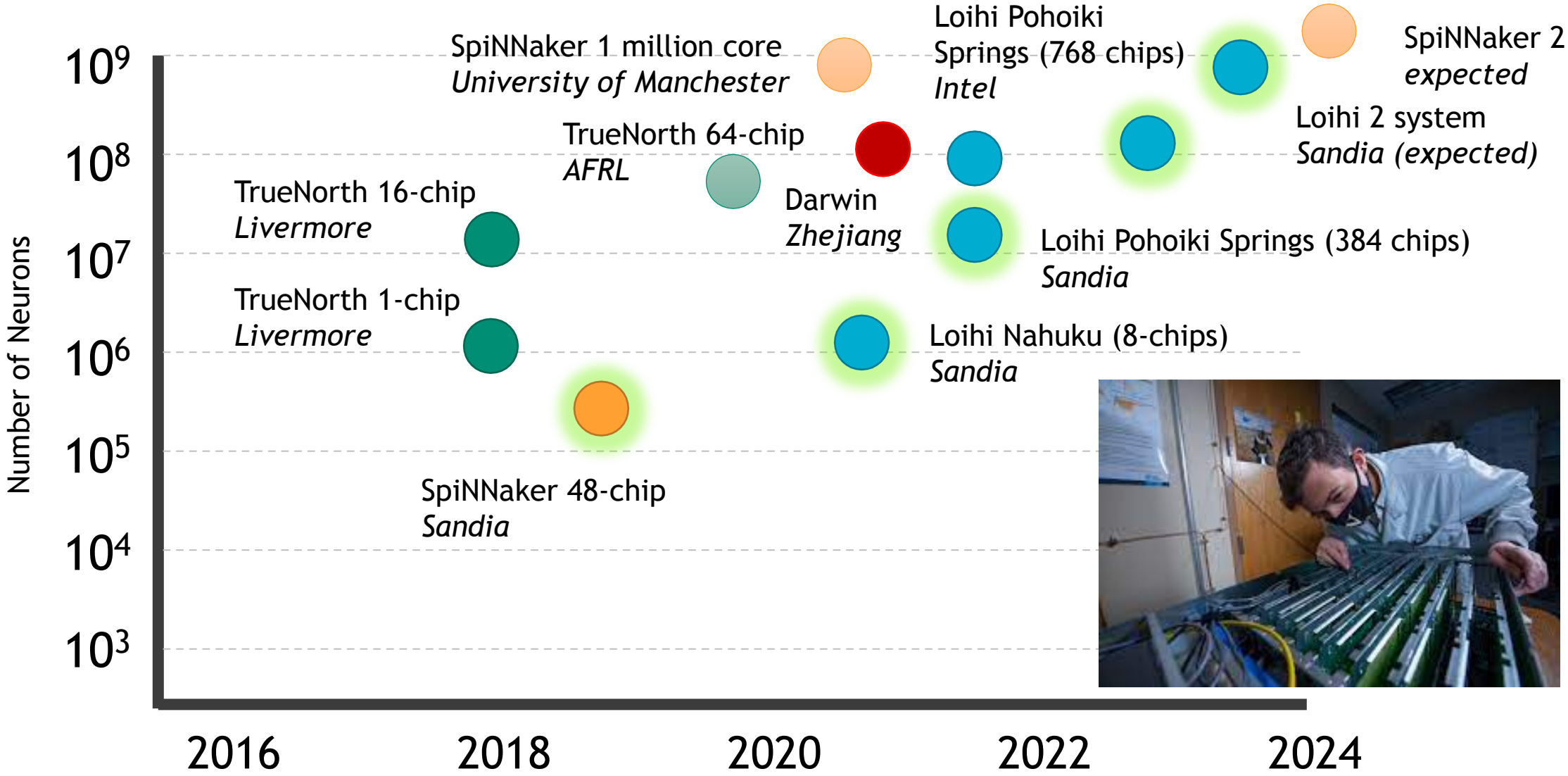


Post-Moore Devices
(ECRAM, Memristors, MTJs, optical, organic, etc)

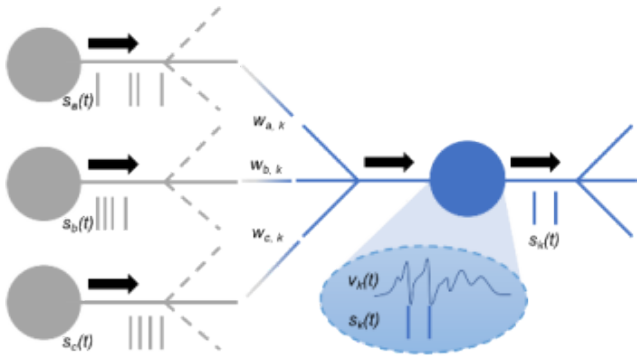
Scale to human sizes?

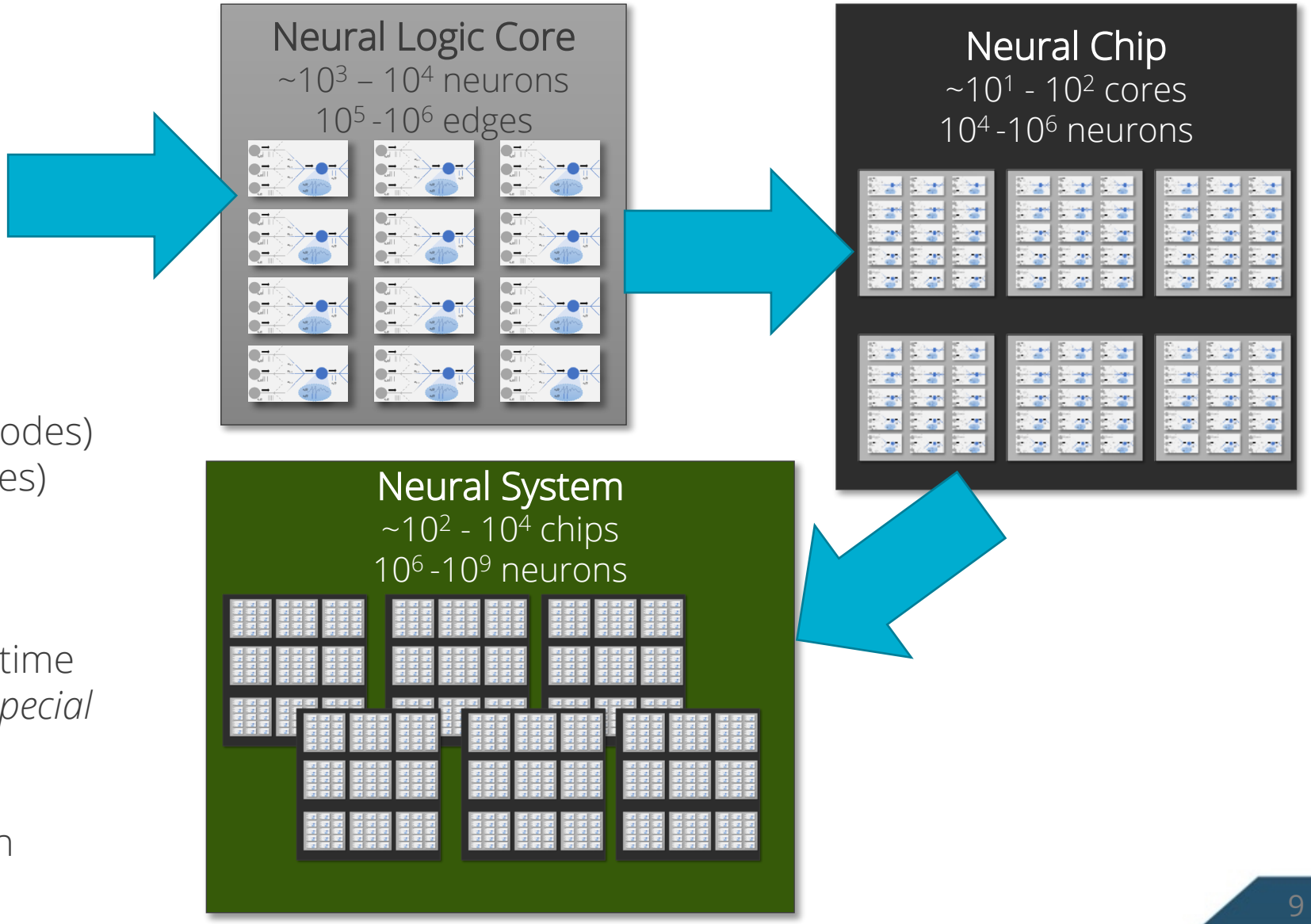
Zidan et al., 2018

Sandia hosts some of the world's largest CMOS-based neuromorphic systems



Spiking neuromorphic today: Overview





Computational Primitives:

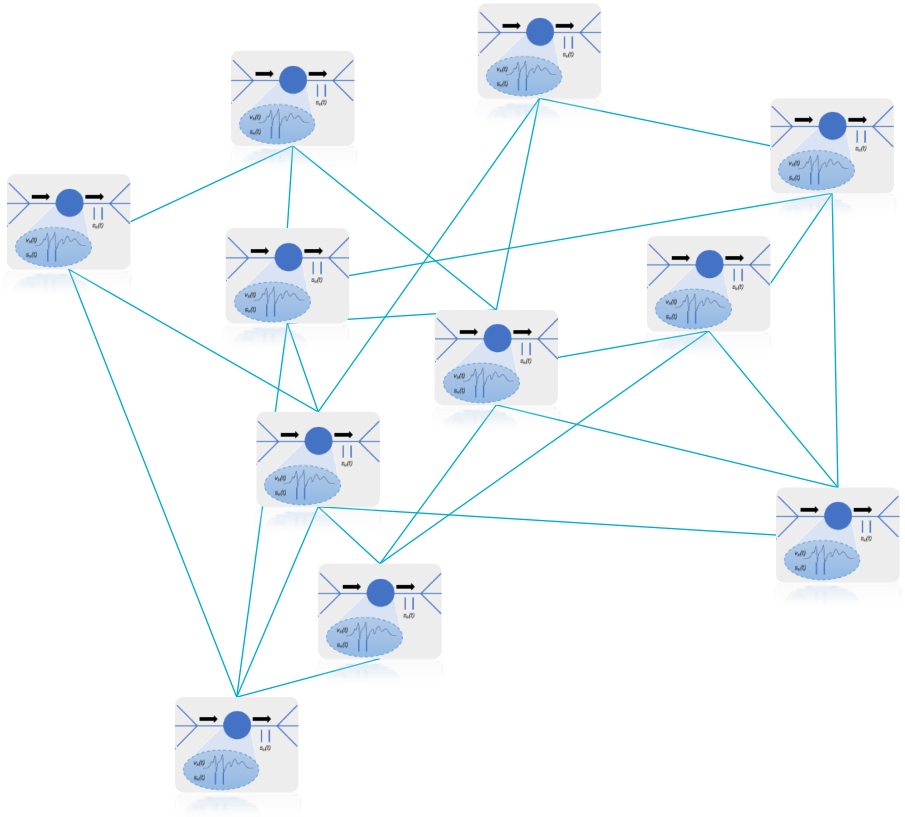
Spiking Neurons (vertices / nodes)

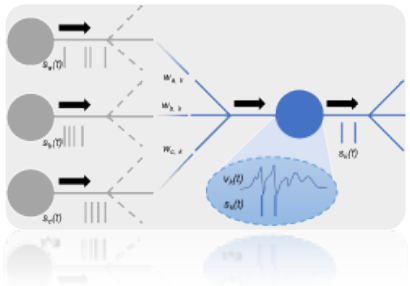
Synapses (connections / edges)

Programmable as arbitrary graphs

- Edges: Directed and weighted
- Nodes: Threshold gate logic + time
- *Artificial neural networks are a special case*
- Programmability, theoretical, analysis and software are open research questions

Neuromorphic hardware jumped ahead of the rest of the stack





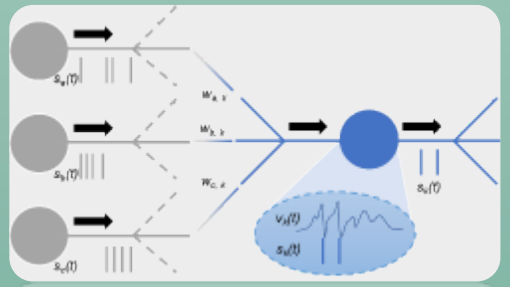
Neuromorphic hardware has been built with a “if we build it, neuroscientists will come” hope

We need

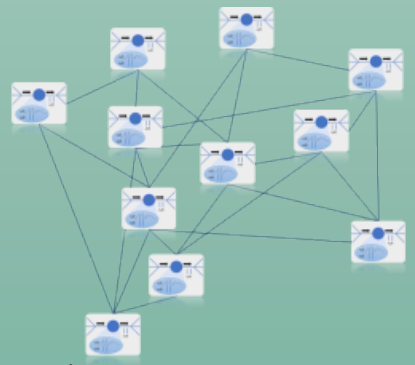
- Driving Applications
- Systems Interface
- Software and Programming Paradigm
- Theoretical Framework

A quick aside: most neuromorphic hardware is ***not*** designed for current artificial neural networks

Neuromorphic Hardware



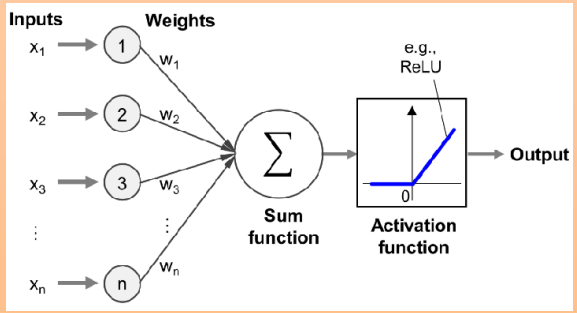
Spiking neurons



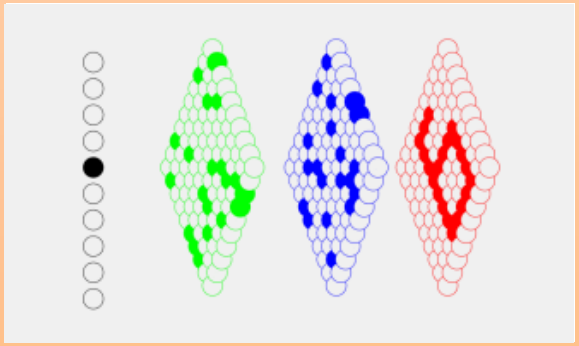
Arbitrary connectivity

- Continual learning integrated into operation
- Inherently temporal
- Dynamical tasks?

Artificial Neural Networks



Continuous neurons

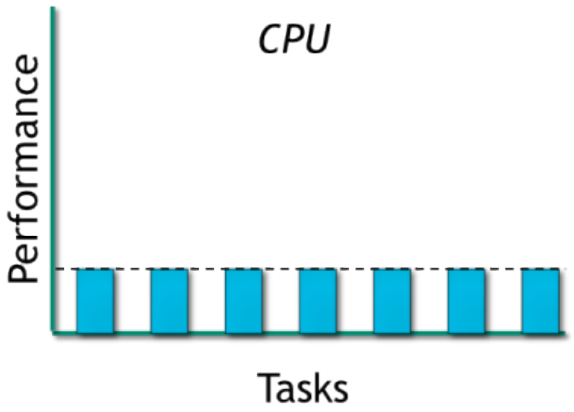


Linear algebra-like networks

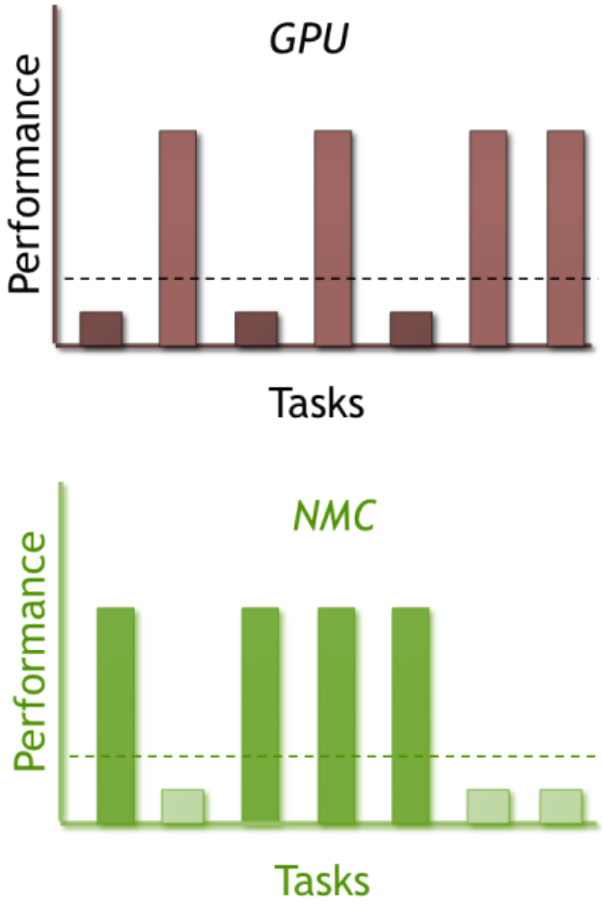
- Distinct training and inference modes
- Time is largely avoided
- Computer vision and natural language

Could neuromorphic be generalized to more algorithms just as GPUs have been?

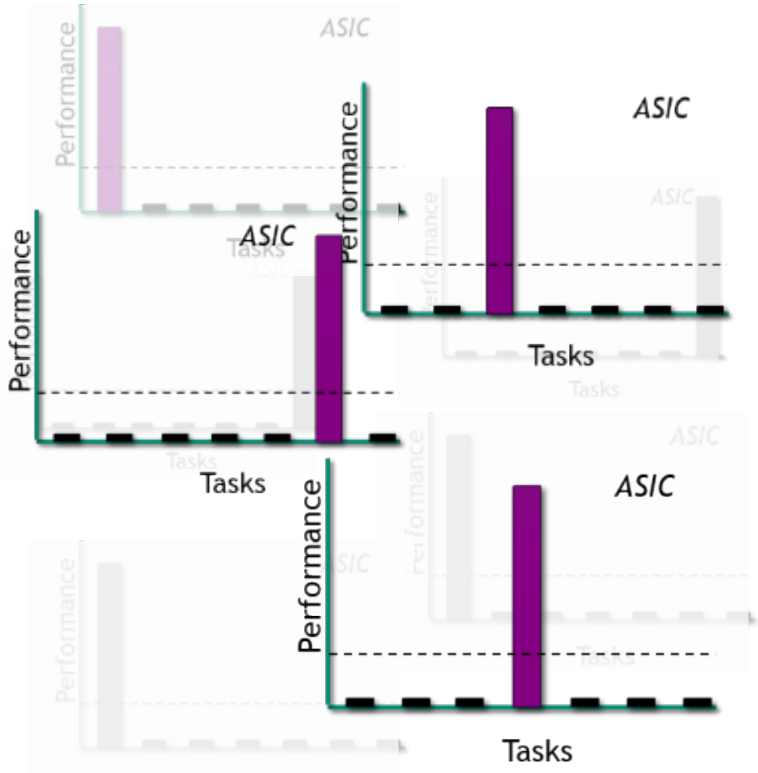
Truly General Purpose



Specialized General Purpose



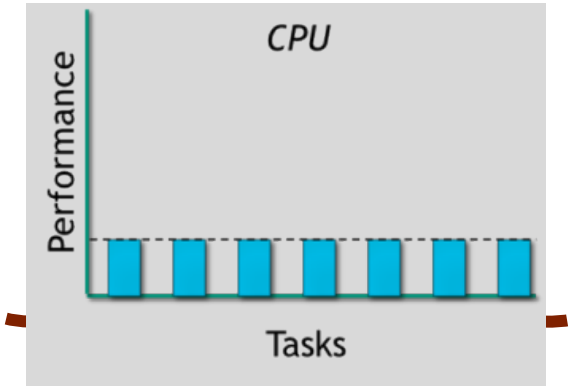
Application Specific



Separating the “can do” from the “should do”

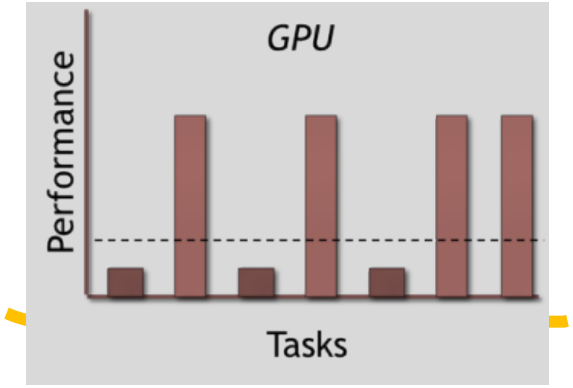
Can implement on NMC, but only to avoid I/O

- Arithmetic (adding, subtraction, multiplication, etc.)
- Data filtering
- Sorting
- Data conversions
- …



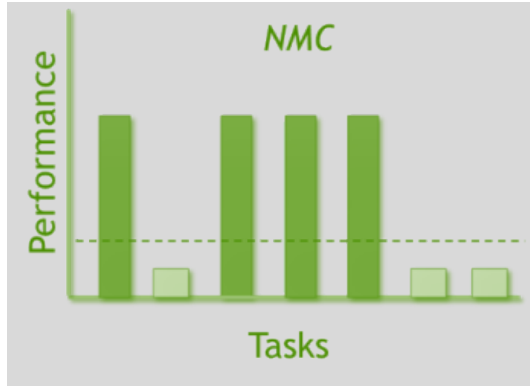
Possibly good on NMC, but there may be alternatives

- Deep learning / conventional artificial neural networks
- Parallel data processing (background and change detection, convolutions, etc)
- Linear algebra (MVM, cross-correlations, L1-norm, etc)
- Classic machine learning (SVMs, k-nearest neighbors, clustering)



Should implement on NMC once systems reach scale

- Algorithms the brain actually uses (* *we don’t have these yet…*)
- Random walks / Discrete Time Monte Carlo
- *Some* Graph Algorithms (Dynamic programming, Djikstra, triangle counting, graph cut, etc)
- *Some* neural networks



Neuromorphic computing can impact a broad range of applications

IOPscience

Neuromorphic Computing and Engineering

ACCEPTED MANUSCRIPT • OPEN ACCESS

A review of non-cognitive applications for neuromorphic computing

James Aimone[1], Prasanna Date[2], Gabriel Fonseca-Guerra[3], Kathleen Hamilton[2], Kyle Henke[4], Bill Kay[5], Garrett Kenyon[4], Shruti Kulkarni[2], Susan Mniszewski[6], Maryam Parsa[7], Sumedh Risbud[3], Catherine Schuman[8], William Severa[1] and J. Darby Smith[1] – Hide full author list

Accepted Manuscript online 10 August 2022 • © 2022 The Author(s). Published by IOP Publishing Ltd

64 Total downloads

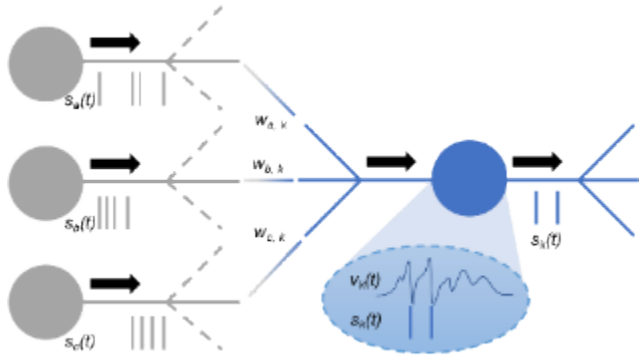
14

Turn on MathJax

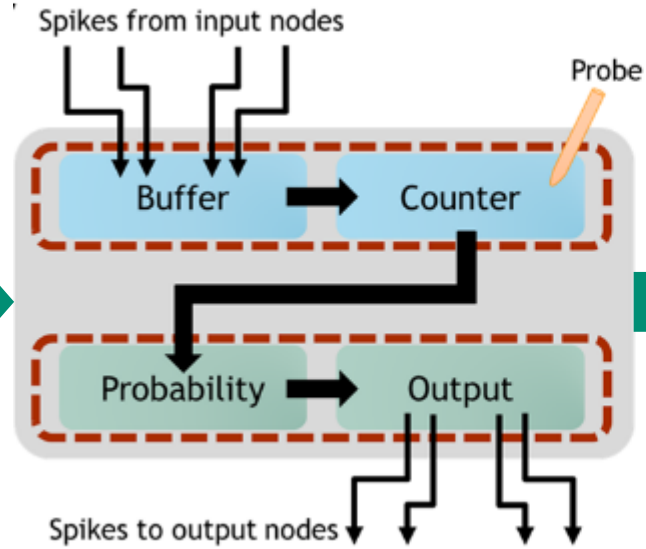
Share this article

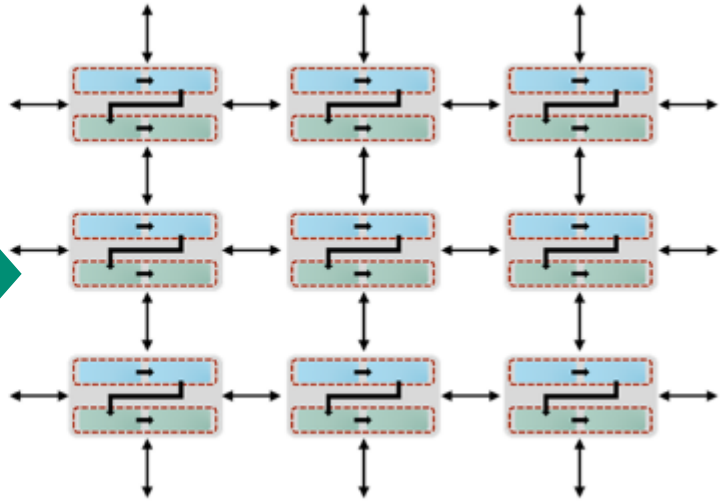
Spiking Scientific Computing

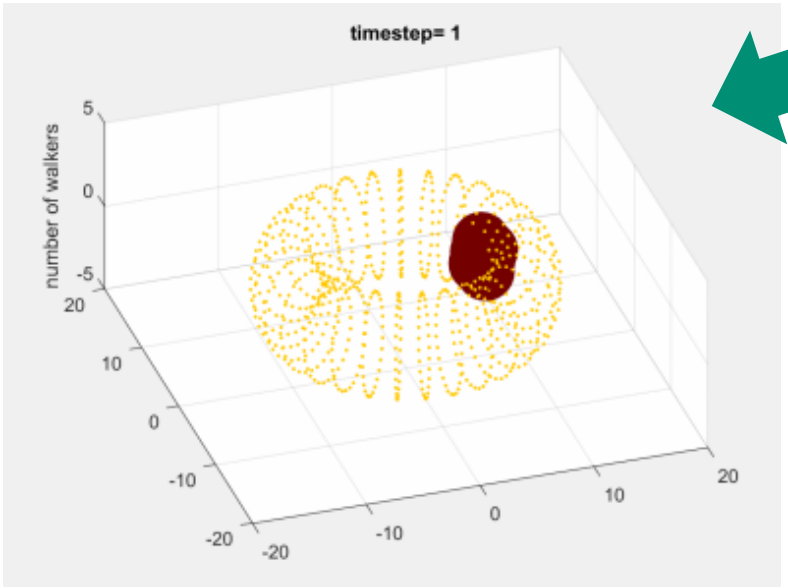
Today's spiking NMC shows energy advantage over conventional approaches on Monte Carlo simulations

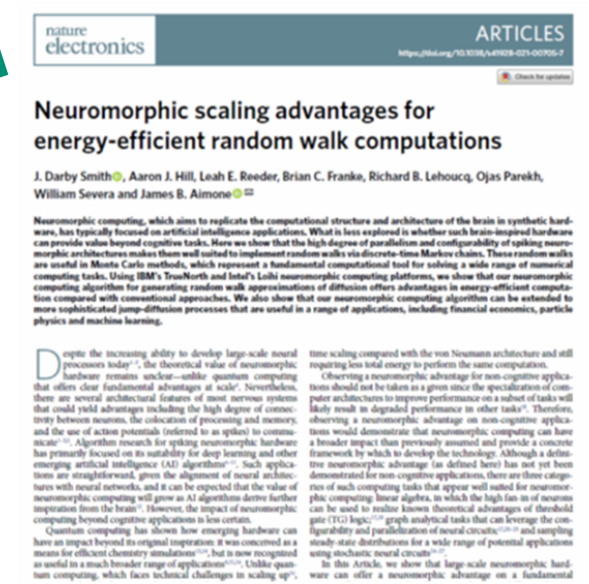


Leaky Integrate and Fire Neuron





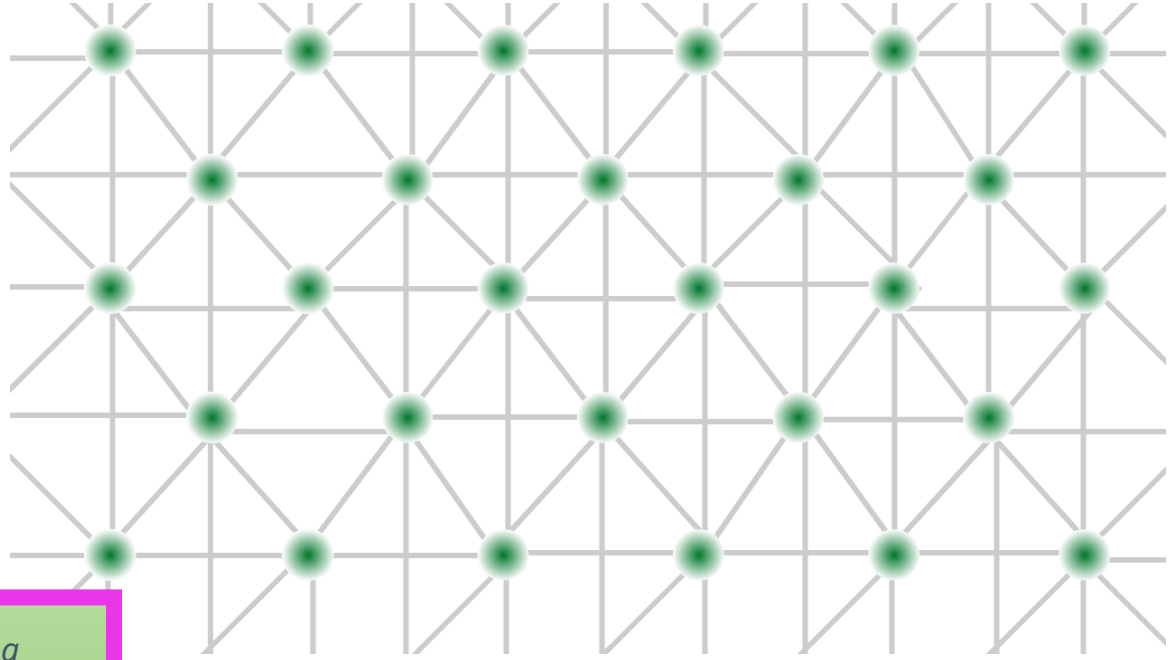




Spiking Scientific Computing

Smith et al., Nature Electronics 2022

Neuromorphic computing advantage appears to be when an algorithm can split task across computational graph with sparse communication



Monte Carlo simulations

Discrete Time Markov Chains

Dynamic programming

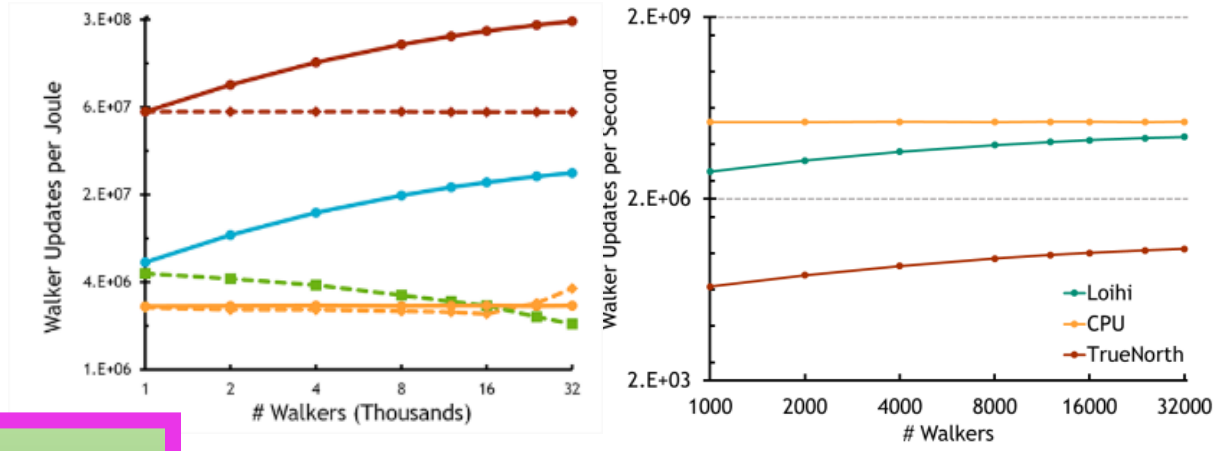
Graph neural networks

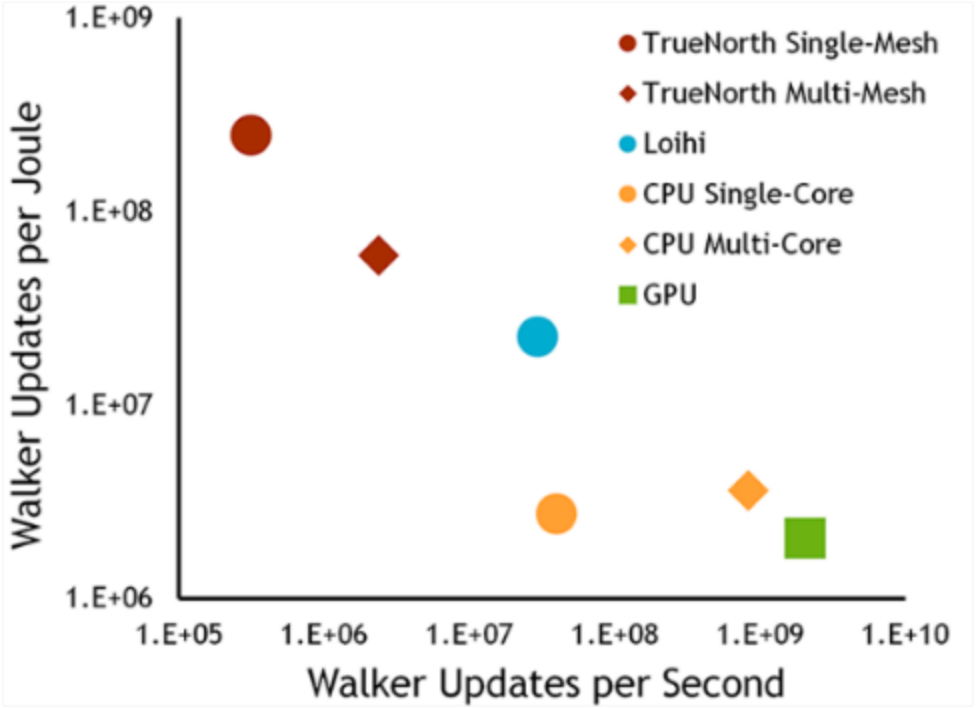
...

Spiking Scientific Computing

We can identify a neuromorphic advantage for simulating random walks

We define a *neuromorphic advantage* as an algorithm that shows a demonstrable **advantage** in terms of one resource (e.g., energy) while exhibiting comparable **scaling** in other resources (e.g., time).





Spiking Scientific Computing

Smith et al., Nature Electronics 2022

Math: What PDEs can these stochastic processes be useful for?

Class of Partial Integro-Differential Equations:

$$\frac{\partial}{\partial t}u(t,\boldsymbol{x}) = \frac{1}{2}\sum_{i,j}(\boldsymbol{a}\boldsymbol{a}^{\top})_{i,j}(t,\boldsymbol{x})\frac{\partial^2}{\partial x_i \partial x_j}u(t,\boldsymbol{x}) + \sum_{i} b_i(t,\boldsymbol{x})\frac{\partial}{\partial x_i}u(t,\boldsymbol{x})$$

$$+\lambda(t,\boldsymbol{x})\int u\left(t,\boldsymbol{x}+\boldsymbol{h}(t,\boldsymbol{x},q)\right) - u(t,\boldsymbol{x})\;\phi_Q(q;t,\boldsymbol{x})\mathrm{d}q$$

$$+c(t,\boldsymbol{x})u(t,\boldsymbol{x}) + f(t,\boldsymbol{x}), \qquad x \in \mathbb{R}^d, t \in [0,\infty).$$

Stochastic Process:

NMC Hardware Simulates This Stochastic Process

$$\mathrm{d}\boldsymbol{X}(t) = \boldsymbol{b}\left(t,\boldsymbol{X}(t)\right)\mathrm{d}t + \boldsymbol{a}\left(t,\boldsymbol{X}(t)\right)\mathrm{d}\boldsymbol{W}(t) + \int \boldsymbol{h}(t,\boldsymbol{X}(t),q)\mathrm{d}P\left(t;Q,\boldsymbol{X}(t)\right).$$

Monte Carlo Approximates This Expectation

Spiking Scientific Computing

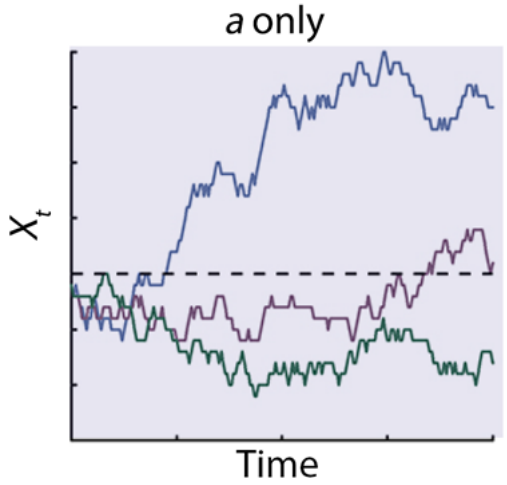
Solution to initial value problem (u(0,x)=g(x)):

$$u(t,\boldsymbol{x}) = \mathbb{E}\left[g\left(\boldsymbol{X}(t)\right)\exp\left(\int_0^t c\left(s,\boldsymbol{X}(s)\right)\mathrm{d}s\right) + \int_0^t f\left(s,\boldsymbol{X}(s)\right)\exp\left(\int_0^s c\left(\ell,\boldsymbol{X}(\ell)\right)\mathrm{d}\ell\right)\mathrm{d}s \;\middle|\; \boldsymbol{X}(0)=\boldsymbol{x}\right].$$

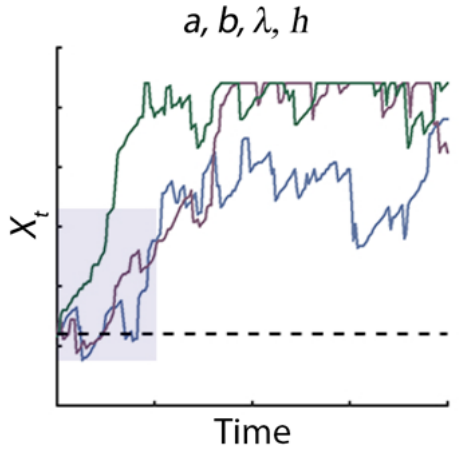
Smith et al., Nature Electronics 2022

Neural MC algorithm can run wide range of stochastic processes

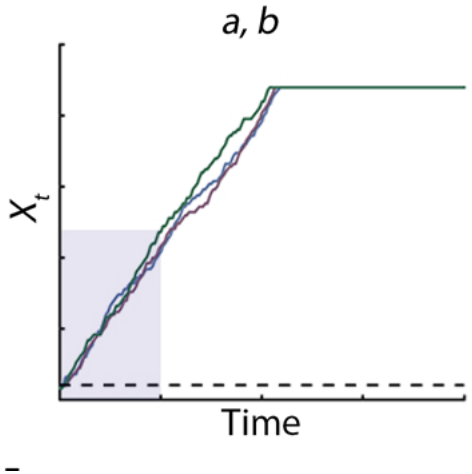
Diffusion



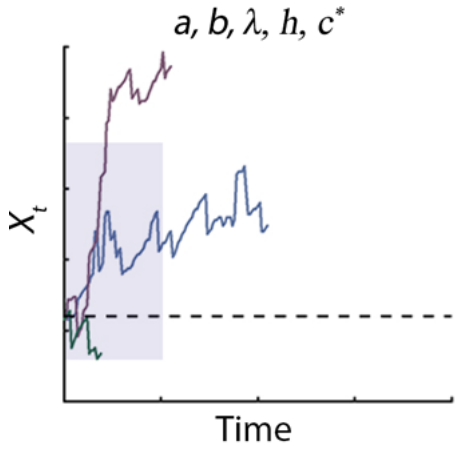
Jump processes



Drift



Absorption / Decay

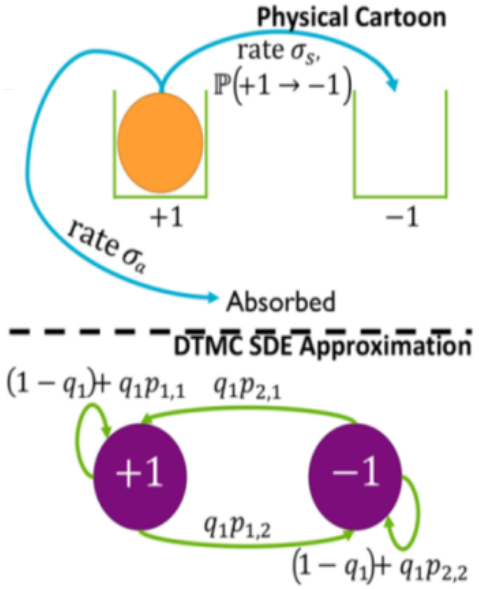


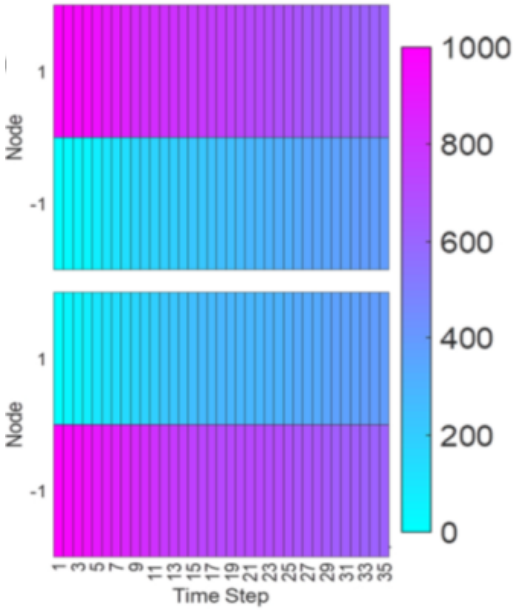
Spiking Scientific Computing

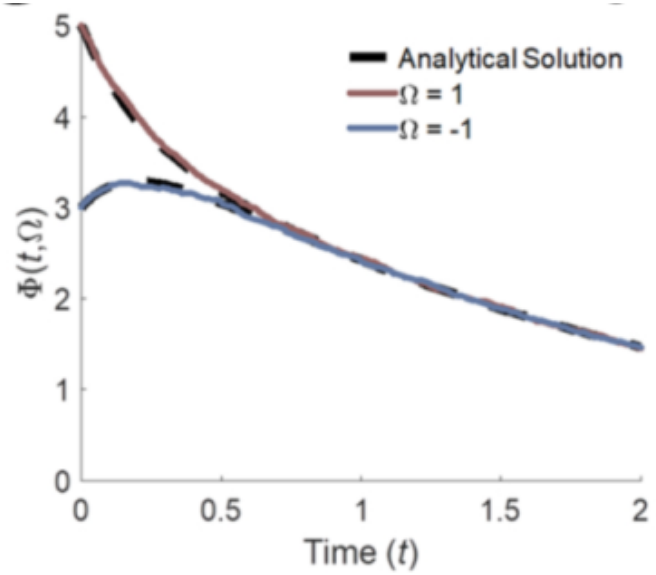
Smith et al., Nature Electronics 2022

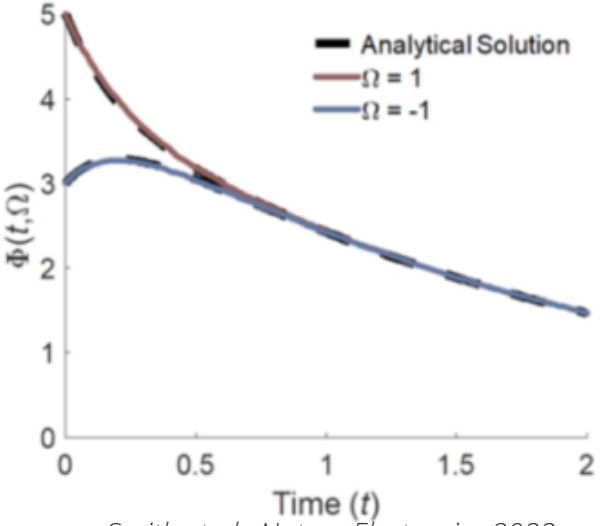
Some more applied examples

- Boltzmann state transition
  - Particle can exist in 2 states (+1 or -1) or be absorbed.
  - Implement as simple stochastic process on TrueNorth







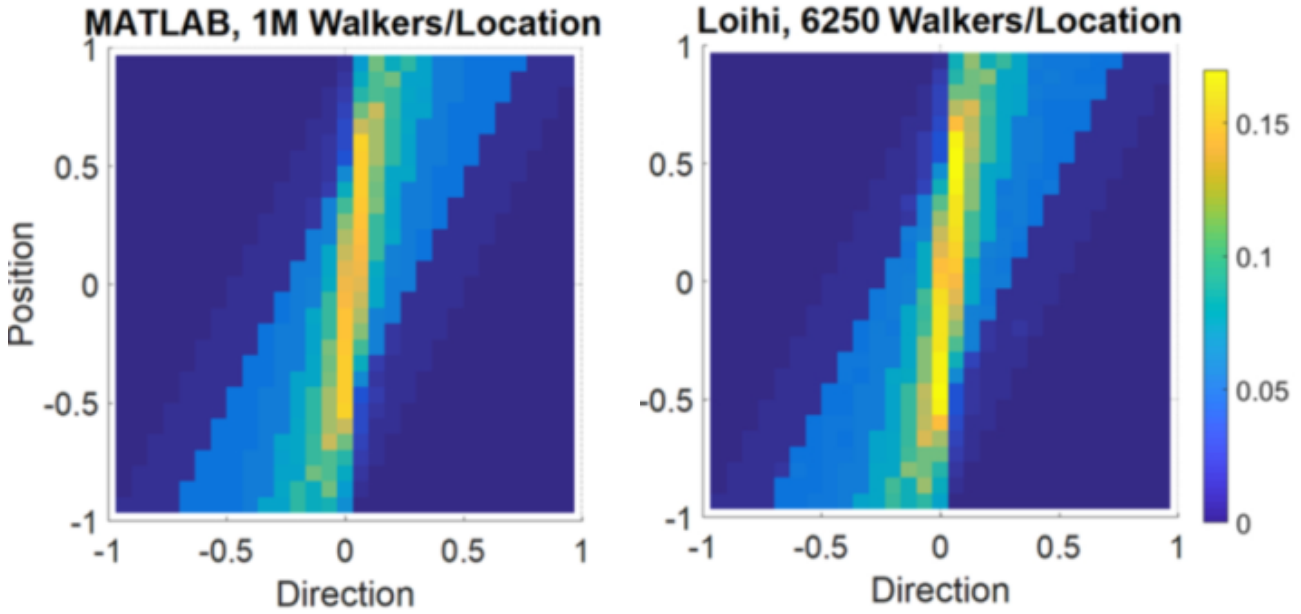


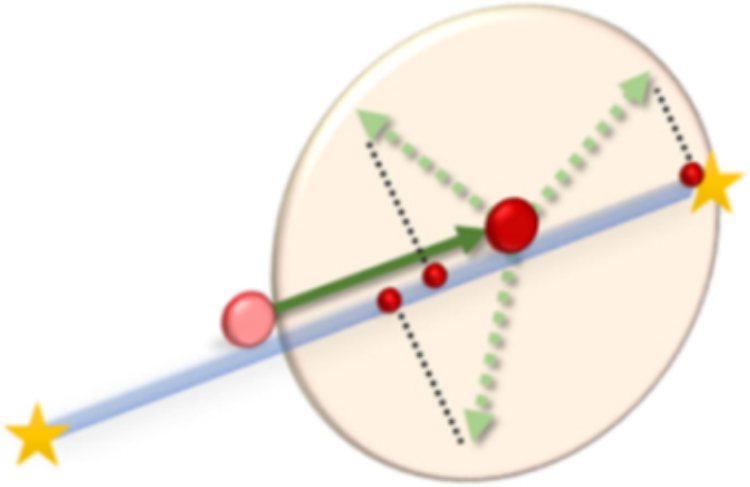
Spiking Scientific Computing

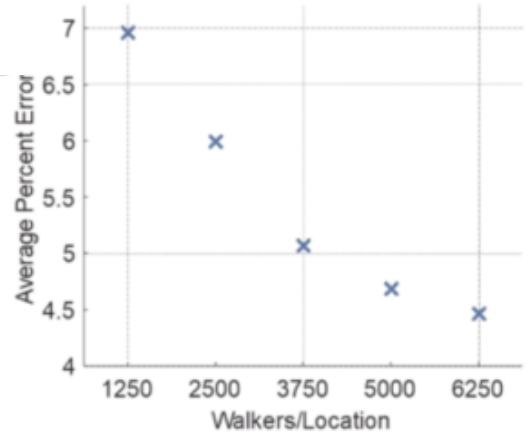
Smith et al., Nature Electronics 2022

Some more applied examples

- 1D particle transport
  - Particle moves in 2D, only track 1D.
  - At point x=0, particle reflects in random direction
  - Track velocity in x-dimension and angle
  - Implemented on Loihi



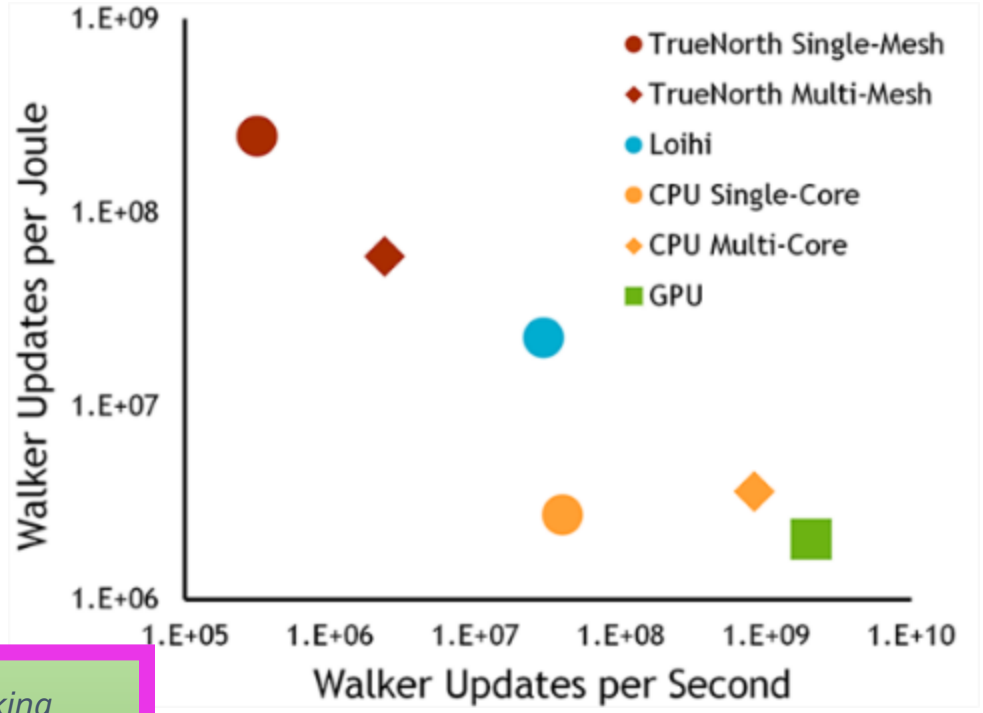




Smith et al., Nature Electronics 2022

Spiking Scientific Computing

Monte Carlo advantage comes from benefits of ***spiking*** … still limited by CMOS



Spiking Scientific Computing

Broad Applications

Monte Carlo simulations

Discrete Time Markov Chains

Graph Analytics

Graph neural networks

…

Limitations

Stochastic sampling

Neuron and synapse scaling

Configurability of neurons

On chip / off chip communication

Today's large scale neuromorphic systems are on ***Pareto Frontier*** of computing

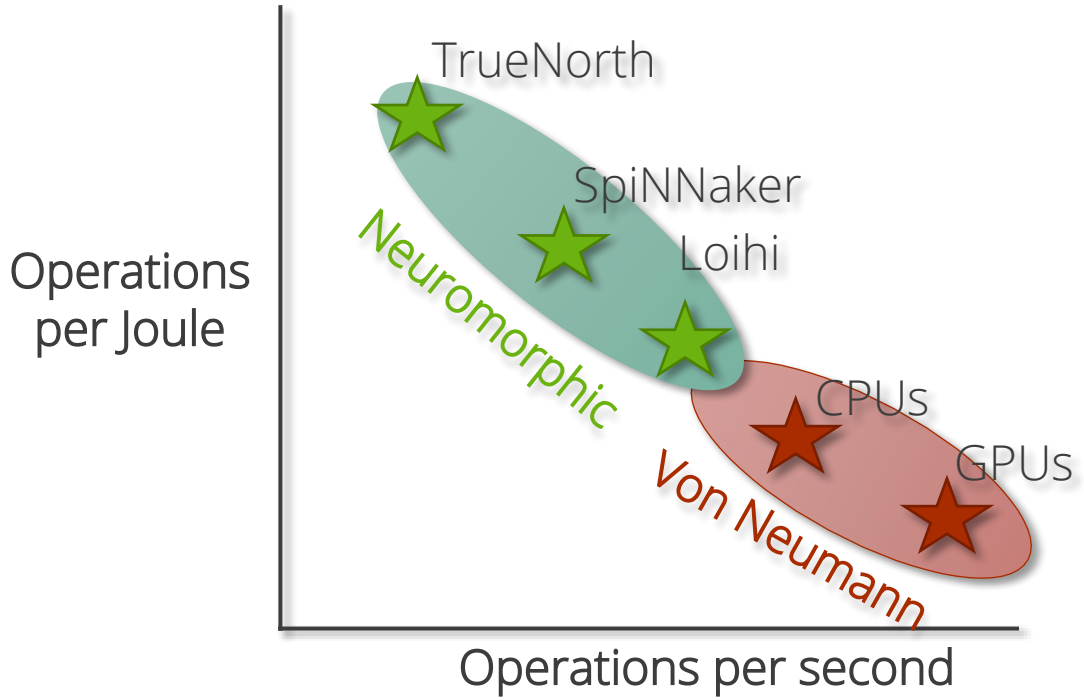
Broad class of algorithms fit this tradeoff
- Monte Carlo / Probabilistic
- Graph analytics
- Artificial intelligence
- Optimization

Architectural advantage
- Event-driven processing
- Massive parallelism

Limitations
- Still CMOS devices
- Architecture is a one time benefit not an extension to Moore's Law

Spiking Scientific Computing

If we're honest; who will pick energy efficiency over speed?

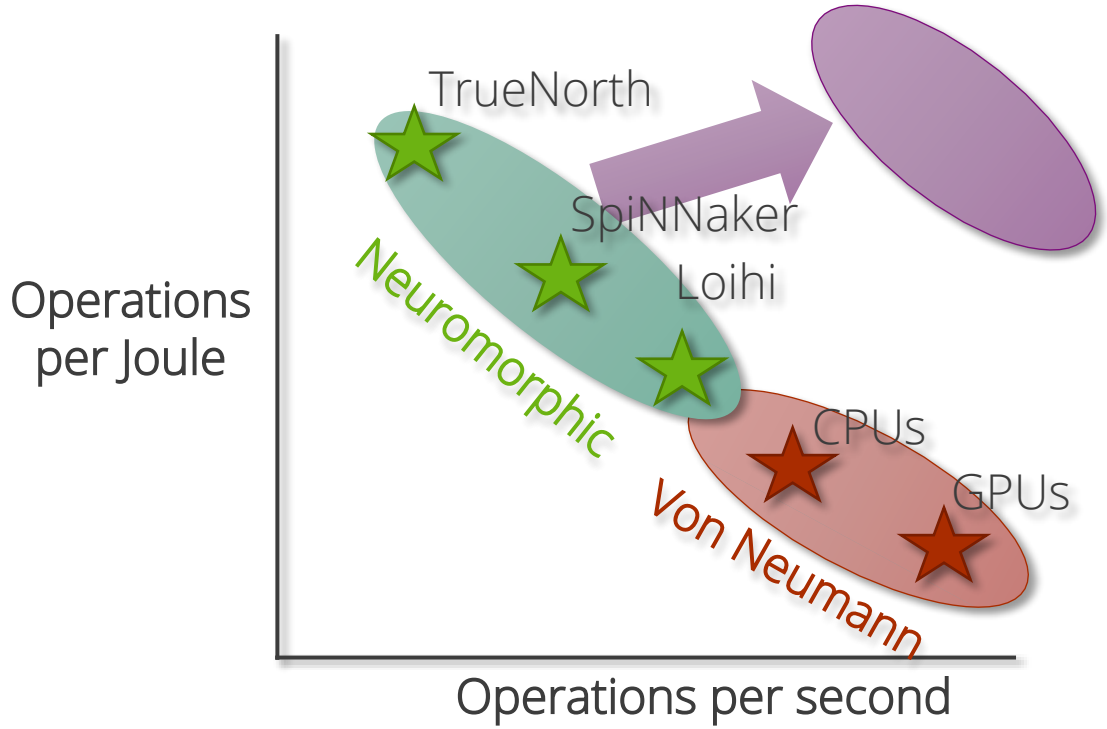


Today's large scale neuromorphic systems are on ***Pareto Frontier*** of computing

Broad class of algorithms fit this tradeoff
- Monte Carlo / Probabilistic
- Graph analytics
- Artificial intelligence
- Optimization

Architectural advantage
- Event-driven processing
- Massive parallelism

Limitations
- Still CMOS devices
- Architecture is a one time benefit not an extension to Moore's Law



Spiking Scientific Computing

Opportunity for Brain-Inspired Materials, Devices & Algorithms

Increasing processing (density, speed, capabilities, etc) while preserving energy advantage and jump neuromorphic over Pareto Frontier

review articles

DOI:10.1145/3231589

Advances in neurotechnologies are reigniting opportunities to bring neural computation insights into broader computing applications.

BY JAMES B. AIMONE

Neural Algorithms and Computing Beyond Moore's Law

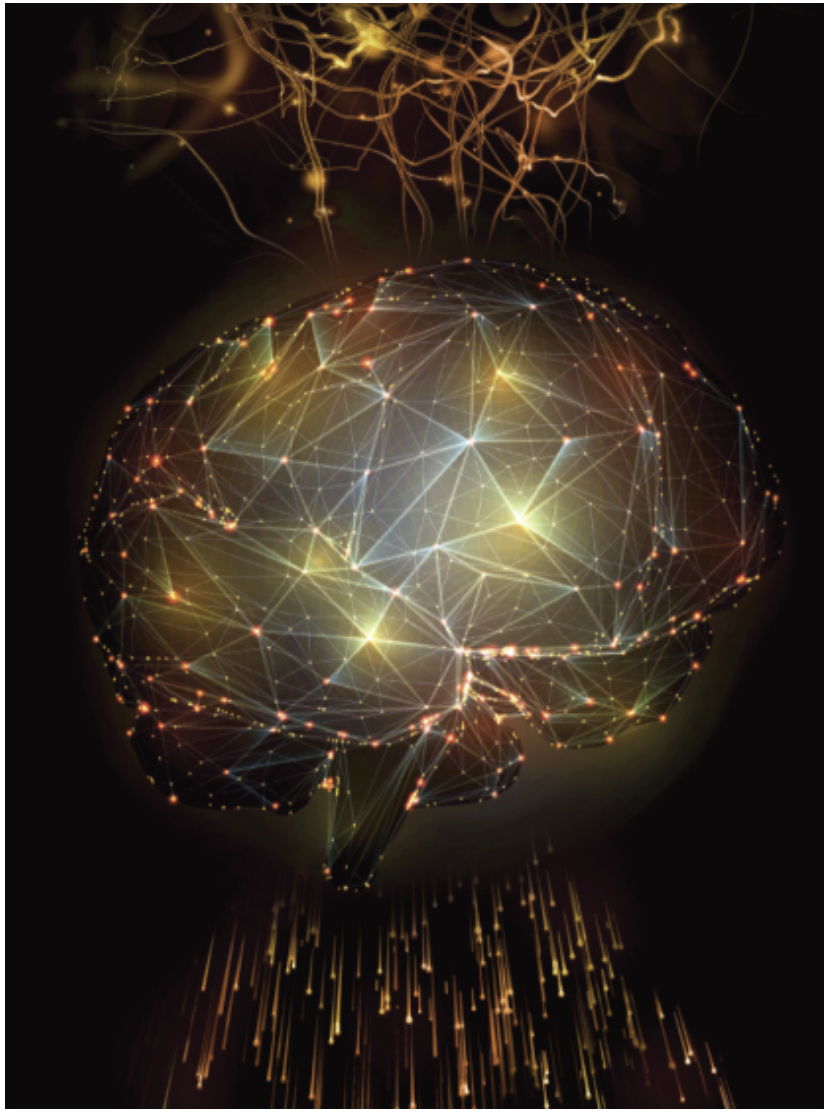
THE IMPENDING DEMISE of Moore's Law has begun to broadly impact the computing research community.[38] Moore's Law has driven the computing industry for many decades, with nearly every aspect of society benefiting from the advance of improved computing processors, sensors, and controllers. Behind these products has been a considerable research industry, with billions of dollars invested in fields ranging from computer science to electrical engineering. Fundamentally, however, the exponential growth in computing described by Moore's Law was driven by advances in materials science.[30,37] From the start, the power of the computer has been limited by the density of transistors. Progressive advances in how to manipulate silicon through advancing lithography methods and new design tools have kept advancing computing in spite of perceived limitations of the dominant fabrication processes of the time.[37]

There is strong evidence that this time is indeed different, and Moore's Law is soon to be over for good.[3,38] Already, Dennard scaling, Moore's Law's lesser known but equally important parallel, appears to have ended.[11] Dennard's scaling refers to the property that the reduction of transistor size came with an equivalent reduction of required power.[3] This has real consequences—even though Moore's Law has continued over the last decade, with feature sizes going from ~65nm to ~10nm; the ability to speed up processors for a constant power cost has stopped. Today's common CPUs are limited to about 4GHz due to heat generation, which is roughly the same as they were 10 years ago. While Moore's Law enables more CPU cores on a chip (and has enabled high power systems such as GPUs to continue advancing), there is increasing appreciation that feature sizes cannot fall much further, with perhaps two or three further generations remaining prior to ending.

Multiple solutions have been presented for technological extension of Moore's Law,[1,5,18,24] but there are two main challenges that must be addressed. For the first time, it is not immediately evident that future materials

» key insights

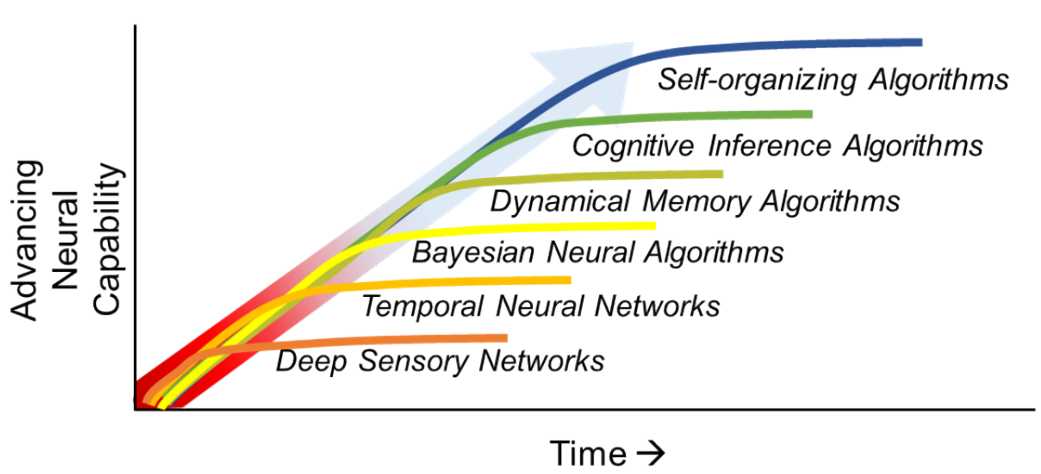
- **While Moore's Law is slowing down, neuroscience is experiencing a revolution, with technology enabling scientists to have more insights into the brain's behavior than ever before and thus positioning the neuroscience field to provide a long-term source of inspiration for novel computing solutions.**
- **Extending the reach of brain-inspiration into computing will not only make current AI methods better, but looking beyond the brain's sensory systems can also expand the reach of AI into new applications.**
- **Realizing the full potential of brain-inspired computing requires increased collaborations and sharing of knowledge between the neuroscience, computer science, and neuromorphic hardware communities.**

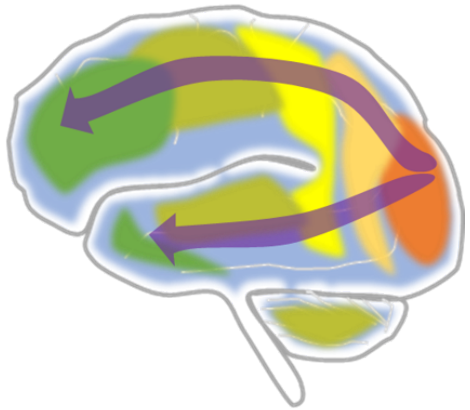


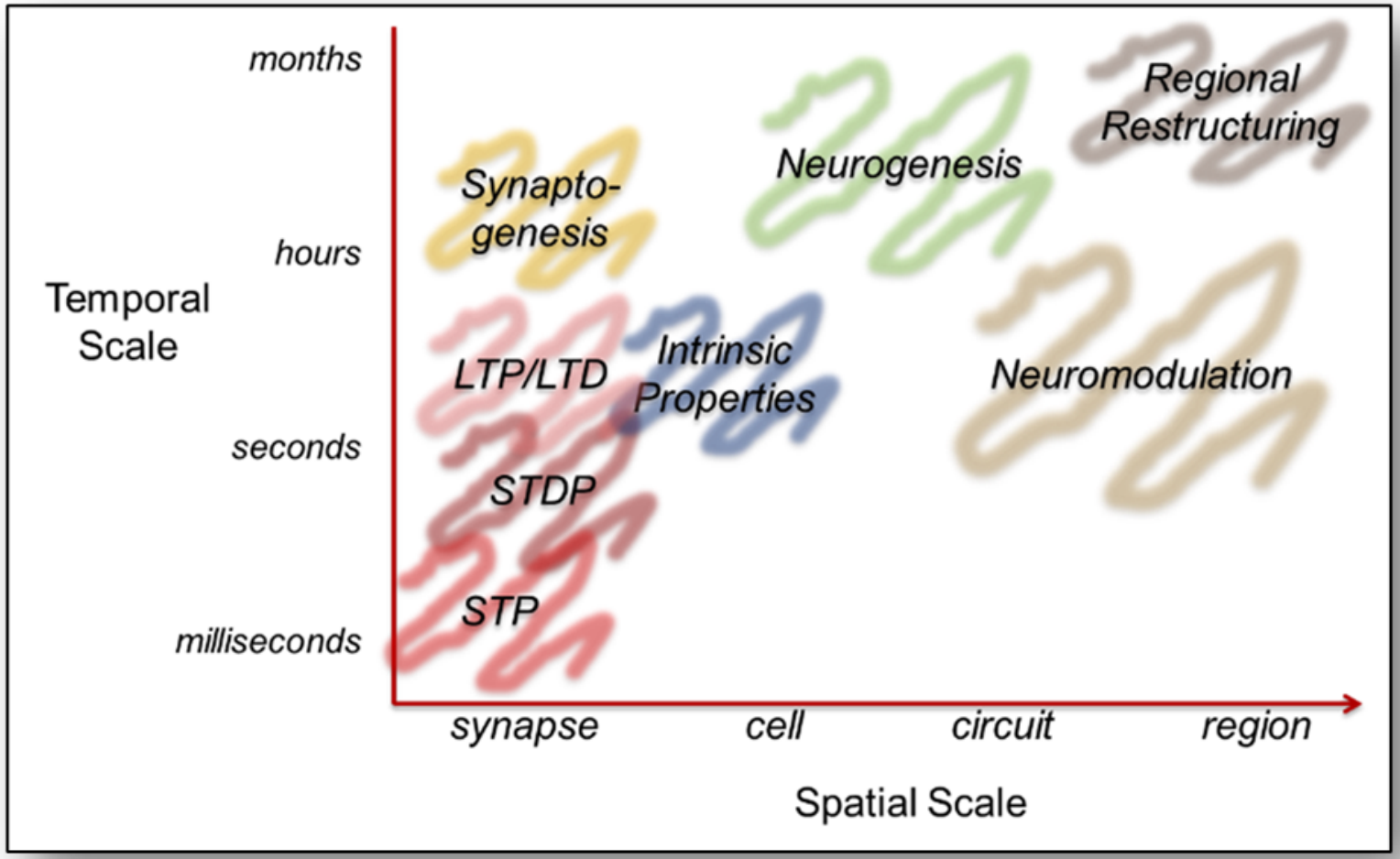
Brain Inspiration

Algorithm Class	Current Algorithms	Inspiration	Application
Deep Vision Processing	Deep Convolutional Networks (VGG, AlexNet, GoogleNet, etc.), HMax, Neocognitron	Hierarchy of sensory nuclei and early sensory cortices	Static feature extraction (e.g., images) & pattern classification
Temporal Neural Networks	Deep Recurrent Networks (long short-term memory), Hopfield Networks	Local recurrence of most biological neural circuits, especially higher sensory cortices	Dynamic feature extraction (e.g., videos, audio) & classification
Bayesian Neural Algorithms	Predictive Coding, Hierarchical Temporal Memory	Substantial reciprocal feedback between "higher" and "lower" sensory cortices	Inference across spatial and temporal scales
Dynamical Memory and Control Algorithms	Liquid State Machines, Echo State Networks, Neural Engineering Framework	Continual dynamics of hippocampus, cerebellum, and prefrontal and motor cortices	Online learning content-addressable memory & adaptive motor control
Cognitive Inference Algorithms	Reinforcement learning (e.g., Q-learning)	Integration of multiple modalities and memory into prefrontal cortex, which provides top-down influence on sensory processing	Context and experience dependent information processing and decision making
Self-organizing Algorithms	Neurogenesis Deep Learning	Initial development and continuous refinement of neural circuits to specific input and outputs	Automated neural algorithm development for unknown input and output transformations

Brain Inspiration







Brain Inspiration

AI systems are slowly starting to incorporate brain-like plasticity

PERSPECTIVE

nature machine intelligence

https://doi.org/10.1038/s42256-022-00452-0

Biological underpinnings for lifelong learning machines

Dhireesha Kudithipudi[1], Mario Aguilar-Simon[2], Jonathan Babb[3], Maxim Bazhenov[4], Douglas Blackiston[5,6], Josh Bongard[7], Andrew P. Brna[2], Suraj Chakravarthi Raja[8], Nick Cheney[7], Jeff Clune[9], Anurag Daram[1], Stefano Fusi[10], Peter Helfer[3], Leslie Kay[11], Nicholas Ketz[12], Zsolt Kira[13], Soheil Kolouri[14], Jeffrey L. Krichmar[15], Sam Kriegman[5,6], Michael Levin[5,6], Sandeep Madireddy[16], Santosh Manicka[8], Ali Marjaninejad[8], Bruce McNaughton[15], Risto Miikkulainen[17], Zaneta Navratilova[15], Tej Pandit[1], Alice Parker[8], Praveen K. Pilly[12], Sebastian Risi[18], Terrence J. Sejnowski[4,19], Andrea Soltoggio[20], Nicholas Soures[1,21], Andreas S. Tolias[22], Darío Urbina-Meléndez[8], Francisco J. Valero-Cuevas[8], Gido M. van de Ven[22], Joshua T. Vogelstein[23], Felix Wang[24], Ron Weiss[3], Angel Yanguas-Gil[16], Xinyun Zou[15] and Hava Siegelmann[25]

Biological organisms learn from interactions with their environment throughout their lifetime. For artificial systems to successfully act and adapt in the real world, it is desirable to similarly be able to learn on a continual basis. This challenge is known as lifelong learning, and remains to a large extent unsolved. In this Perspective article, we identify a set of key capabilities that artificial systems will need to achieve lifelong learning. We describe a number of biological mechanisms, both neuronal and non-neuronal, that help explain how organisms solve these challenges, and present examples of biologically inspired models and biologically plausible mechanisms that have been applied to artificial systems in the quest towards development of lifelong learning machines. We discuss opportunities to further our understanding and advance the state of the art in lifelong learning, aiming to bridge the gap between natural and artificial intelligence.

Learning is a defining ability of biological systems, whereby experience leads to behavioural adaptations that improve performance[1]. The past couple of decades have witnessed astonishing advances in the field of machine learning. Nevertheless, a new generation of applications—self-driving cars and trucks, autonomous drones, delivery robots, intelligent handheld and wearable devices, and others that we have not yet imagined—will require a new type of machine intelligence that is able to learn throughout its lifetime. Such machines will need to acquire new skills without compromising old ones, adapt to changes, and apply previously learned knowledge to new tasks—all while conserving limited resources such as computing power, memory and energy. These capabilities are collectively known as lifelong learning (L2).

In contrast to the current generation of intelligent machines, animal species ranging from invertebrates to humans are able to learn continually throughout their lifetime. Neuroscientists and other biologists have proposed several mechanisms to explain this ability, and machine learning researchers have attempted to emulate them in artificial systems, with varying degrees of success. In this Perspective article, we examine our current understanding of how biological organisms learn continually and review the state of the art in biologically inspired L2 models. We describe a variety of biological mechanisms, both neuronal and non-neuronal, that can improve our ability to create highly functioning lifelong learning machines.

It should be noted that there is also a body of artificial intelligence (AI) research that tackles the lifelong learning problem from a less clearly biological perspective[2–7]. These can be broadly organized into three types: 'rehearsal', which store or generate data from past tasks for replay[8–11]; 'architectural', which expand the model parameters[12–15]; and 'regularization-based' approaches, which penalize changes to parameters important to past tasks[16–19] or use meta-learning[20]. Such models, which are not directly inspired by a biological mechanism, fall outside the scope of this Perspective.

In this Perspective, we will (1) identify a set of key features of lifelong learning; (2) provide an overview of biological mechanisms that are believed to be involved in realizing these features; and (3) review research in which analogous mechanisms have been implemented in machine learning models with the aim of realizing lifelong learning capabilities in artificial systems. We conclude with a look at future challenges and opportunities.

[1]University of Texas at San Antonio, San Antonio, TX, USA. [2]Intelligent Systems Laboratory, Teledyne Scientific, RTP, NC, USA. [3]Massachusetts Institute of Technology, Boston, MA, USA. [4]University of California at San Diego, La Jolla, CA, USA. [5]Allen Discovery Center, Tufts University, Medford, MA, USA. [6]Wyss Institute, Harvard University, Cambridge, MA, USA. [7]University of Vermont, Burlington, VT, USA. [8]University of Southern California, Los Angeles, CA, USA. [9]University of British Columbia, Vancouver, BC, Canada. [10]Columbia University, New York, NY, USA. [11]University of Chicago, Chicago, IL, USA. [12]HRL Laboratories, Malibu, CA, USA. [13]Georgia Institute of Technology, Atlanta, GA, USA. [14]Vanderbilt University, Nashville, TN, USA. [15]University of California, Irvine, CA, USA. [16]Argonne National Laboratory, Lemont, IL, USA. [17]The University of Texas at Austin, Austin, TX, USA. [18]IT University of Copenhagen, Copenhagen, Denmark. [19]Salk Institute for Biological Studies, La Jolla, CA, USA. [20]Loughborough University, Loughborough, UK. [21]Rochester Institute of Technology, Rochester, NY, USA. [22]Baylor College of Medicine, Houston, TX, USA. [23]Johns Hopkins University, Baltimore, MA, USA. [24]Sandia National Laboratories, Albuquerque, NM, USA. [25]University of Massachusetts, Amherst, MA, USA. ✉e-mail: dk@utsa.edu

NATURE MACHINE INTELLIGENCE | VOL 4 | MARCH 2022 | 196–210 | www.nature.com/naturemachineintell

		Key features					
		Transfer and adaptation	Overcoming catastrophic forgetting	Exploiting task similarity	Task-agnostic learning	Noise tolerance	Resource efficiency and sustainability
Biologically inspired mechanisms	Neurogenesis		169–174	234	161		174,201,202
	Episodic replay		54,175,176,179, 180		54,176	176,177	53,54,175,176, 179,180,203
	Metaplasticity		67,89,181–185		7,181,185,198		89,181–183,198
	Neuromodulation	70,78,84–86,88, 89,157,159,160	78,79,84,89,164	89	78,159	78,158,199	89
	Context-dependent perception and gating	78,79,158,161–167	78,168	79,162–166	70,161	158,162,163	
	Hierarchical distributed systems			188–191		113,191,200	191
	Cognition outside the brain			195–197			
	Reconfigurable organisms	139		139,147		139,147	139,147
	Multisensory integration			152,155,192,193		113,162	

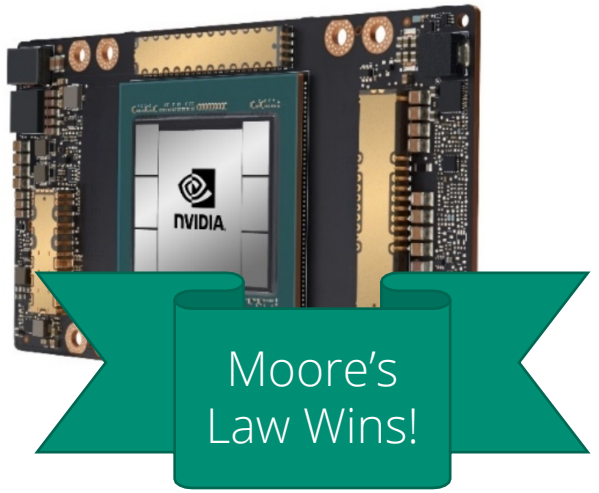
Brain Inspiration

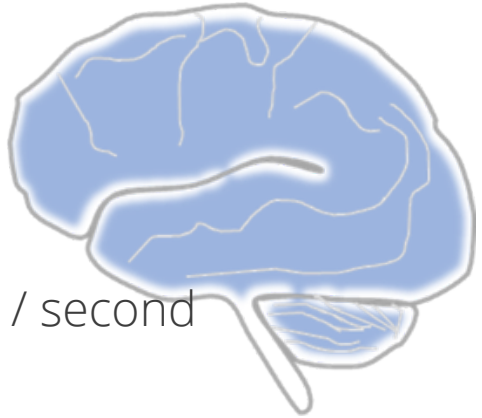
A concrete future direction: Brain-inspired systems may need to embrace stochasticity

Has the tremendous success of deterministic computing left probabilistic applications behind?

Which approach is best to interpret a clear input?

~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic



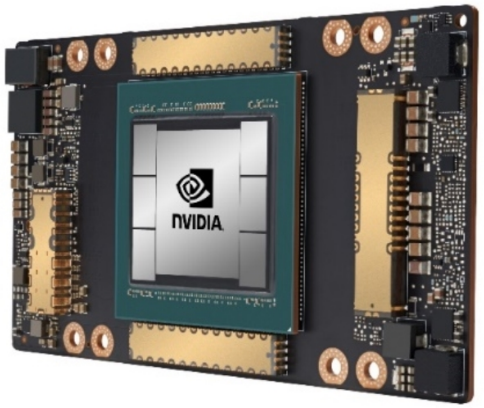


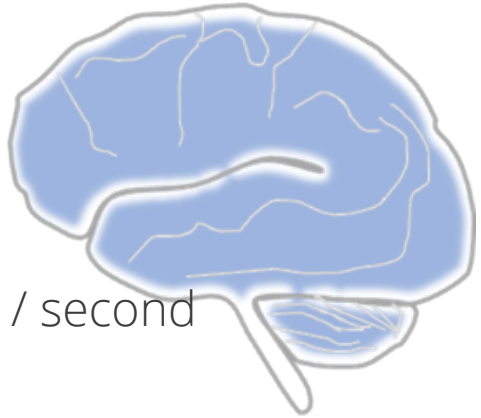
~20 W
~10^{15} synaptic events / second
Fully stochastic

Has the tremendous success of deterministic computing left probabilistic applications behind?

Which approach is best to interpret an ambiguous input?

~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic



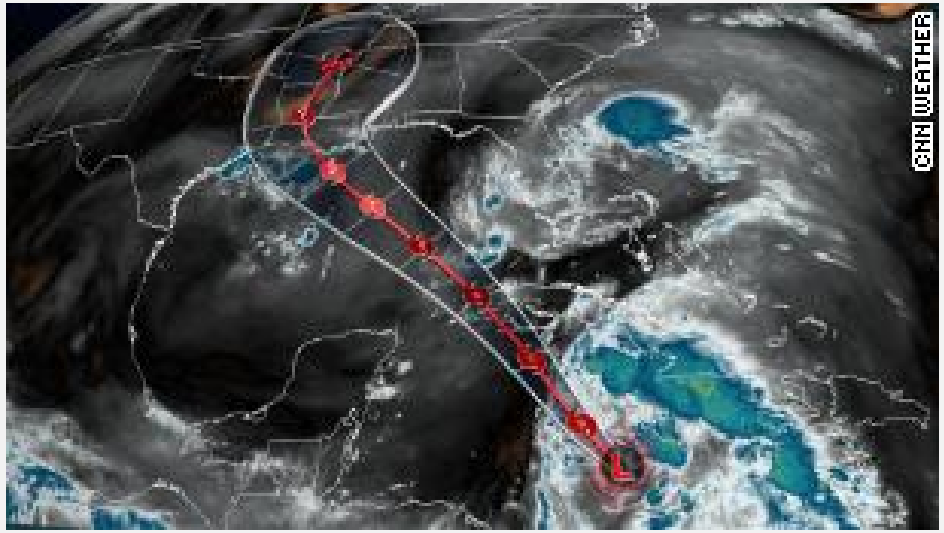


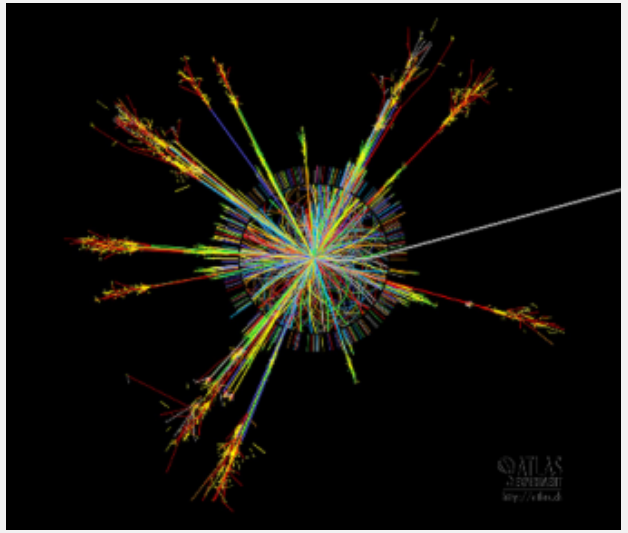
~20 W
~10^{15} synaptic events / second
Fully stochastic

Computing applications face challenges in uncertainty

Artificial Intelligence

- Bayesian neural networks are appealing yet often computationally intractable



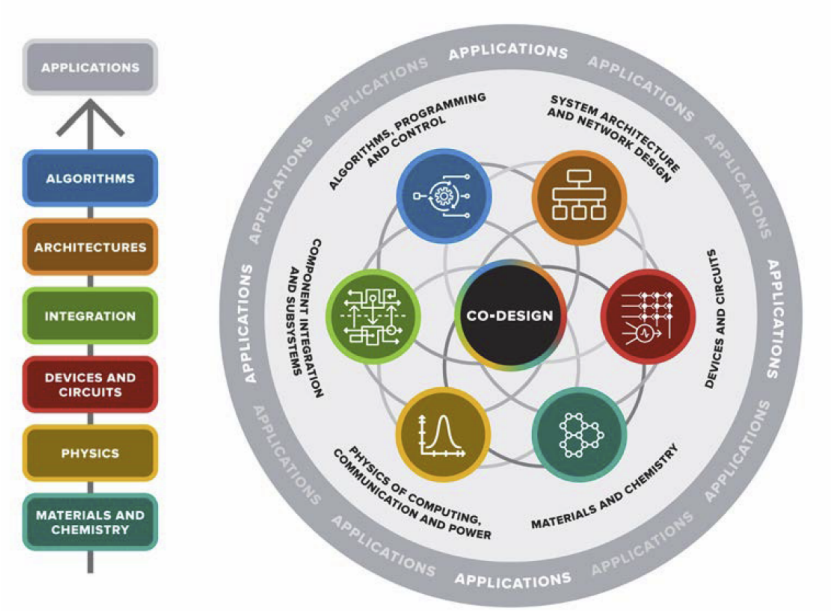


Modeling and Simulation

- Modeling uncertainties is critical in the use of even fully deterministic simulations
- Many applications are inherently stochastic in their physics and are best modeled using probabilistic methods

CO-designed Improved Neural Foundations Leveraging Inherent Physics Stochasticity (COINFLIPS)

- Office of Science Co-Design in Microelectronics program
  - Co-funded through ASCR and BES, participation by NP, HEP, and FES



*To enable new generations of energy-efficient computing systems over the next decade, **a complete reconceptualization of the science and technology underlying the microelectronics co-design approach is needed** to integrate emerging devices, materials, interconnects, and non-linear phenomena with the needs of scientific computing applications.*

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories

NYU

The University of Texas at Austin

UT

OAK RIDGE National Laboratory

CO-designed Improved Neural Foundations Leveraging Inherent Physics Stochasticity (COINFLIPS)

- Office of Science Co-Design in Microelectronics program
  - Co-funded through ASCR and BES, participation by NP, HEP, and FES
- ~COINFLIPS is partnering with a growing number of organizations
  - Andy Kent @ New York University
  - Jean Anne Incorvia @ University of Texas Austin
  - Katie Schuman @ University of Tennessee
  - Prasanna Date @ Oak Ridge National Laboratory
  - Les Bland @ Temple University

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories

NYU

The University of Texas at Austin

UT

OAK RIDGE National Laboratory

We are benefitting from 70 years of microelectronics that embrace ***deterministic*** components to solve ***deterministic*** problems

COINFLIPS sees an opportunity to embrace ***stochastic*** computing to solve ***uncertainty*** problems

Today's computers emulate uncertainty by using pseudo-random number generation

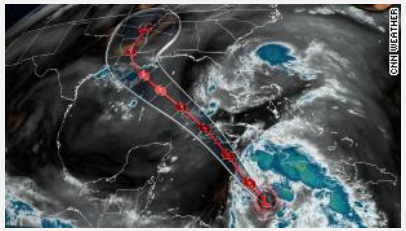
"Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin."

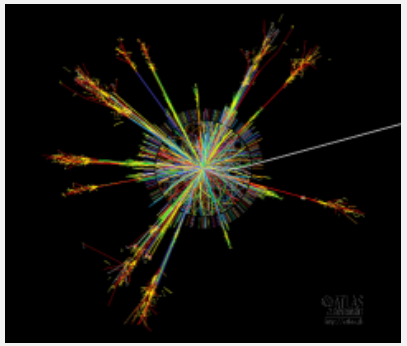
John von Neumann, 1951

70 years later…

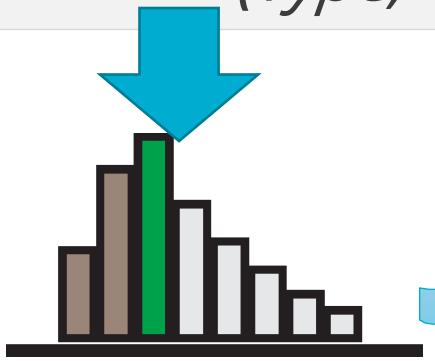
- Pseudo-RNGs can be quite effective, and do offer some advantages in verification, etc.
- But they are expensive, and when they go wrong the implications can be disastrous

COINFLIPS aims to integrate true random number generators using stochastic devices into neuromorphic architectures



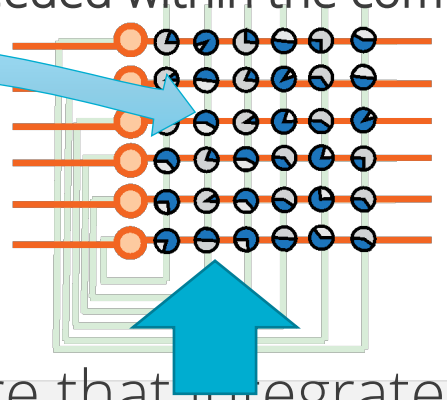


Improved Random Number Generation
(Type, Quantity, Quality)



Sample a random number from the *exact* distribution we require

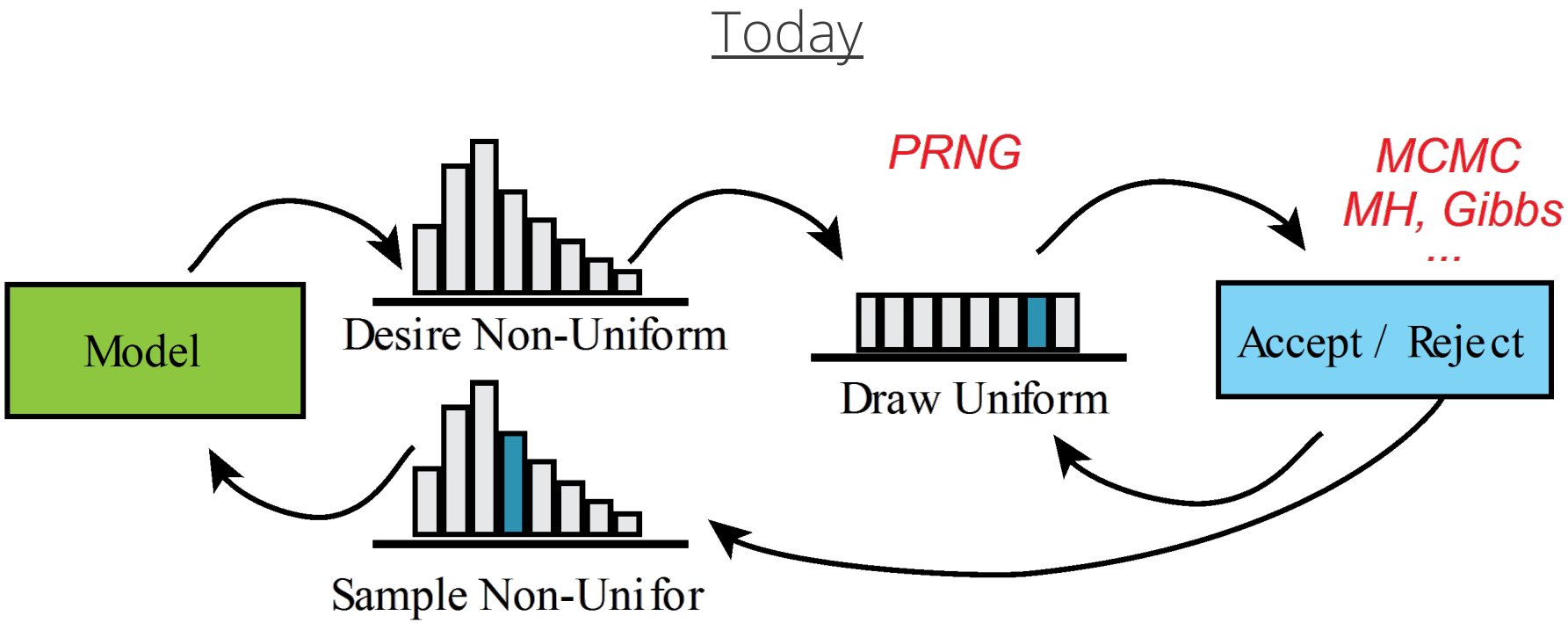
And sample that number *where* it is needed within the computation



Neuromorphic architecture that integrates ubiquitous stochastic devices with computing and memory

COINFLIPS aims to improve both speed and energy of probabilistic computing applications

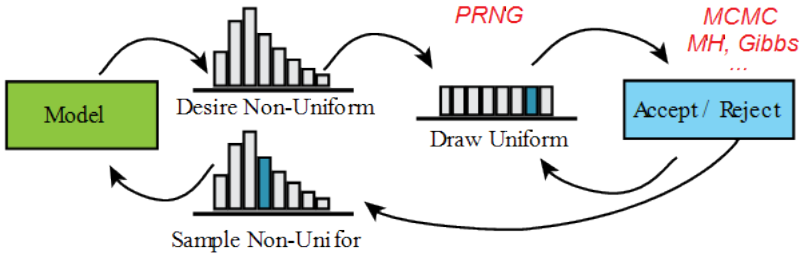
Evaluate opportunity of a probabilistic computing paradigm



COINFLIPS

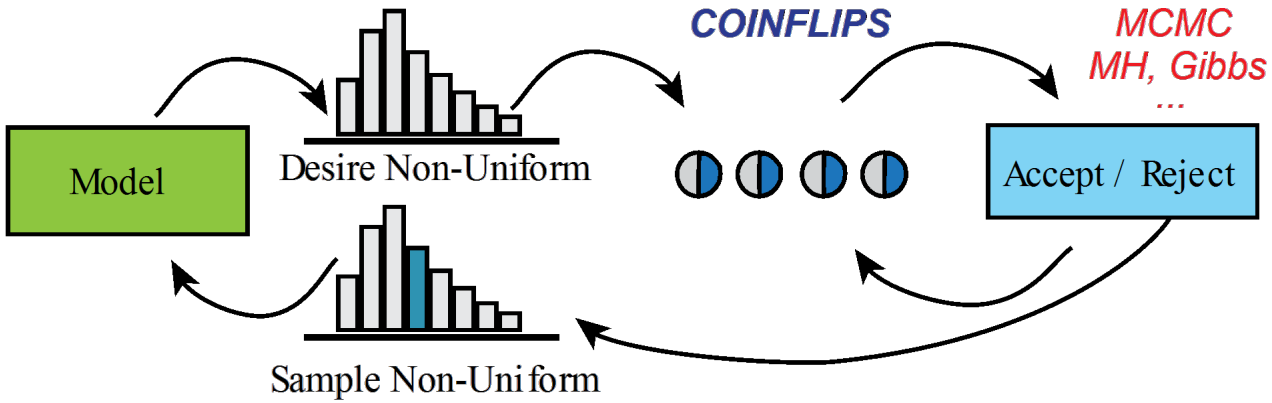
Evaluate opportunity of a probabilistic computing paradigm

Today



COINFLIPS

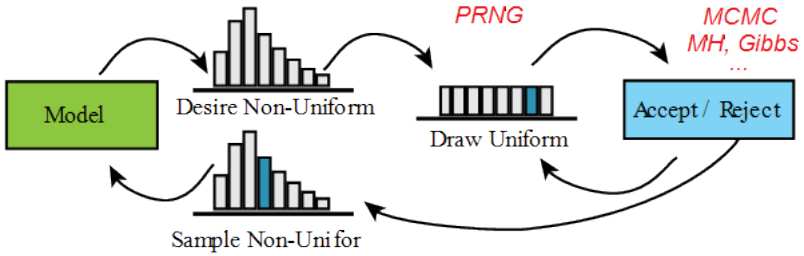
Future?



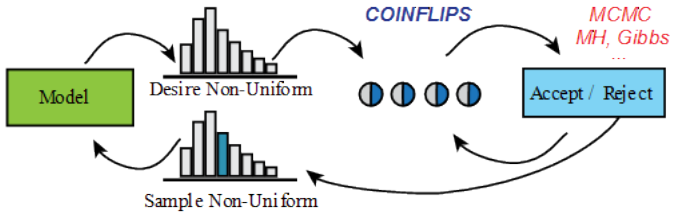
Step 1: Draw suitable uniform RNs from hardware

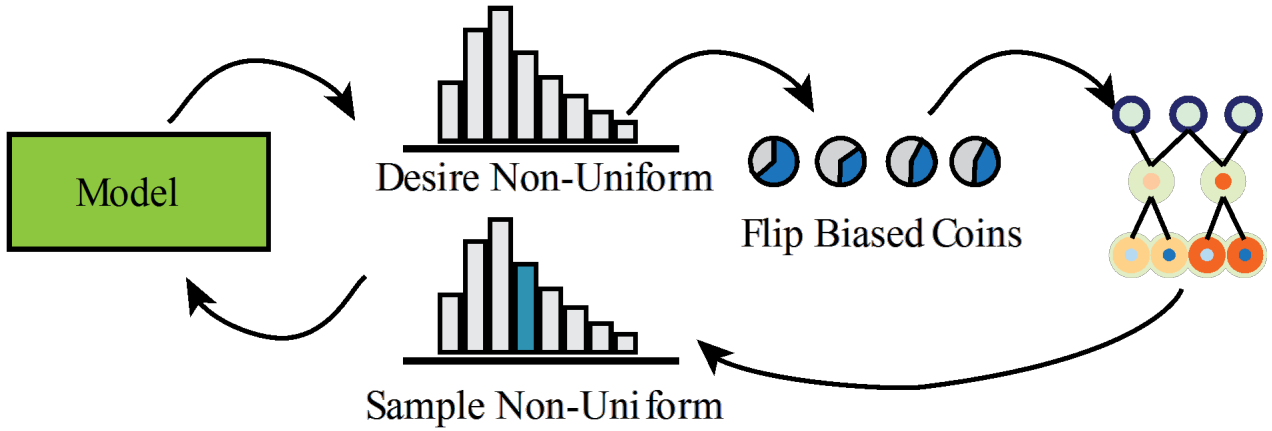
Evaluate opportunity of a probabilistic computing paradigm

Today



Future?



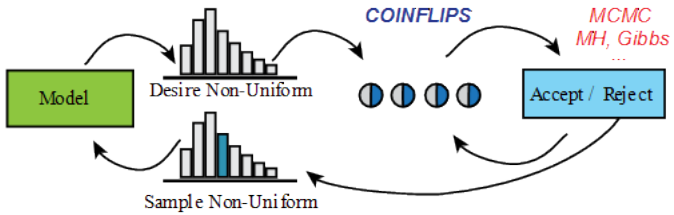


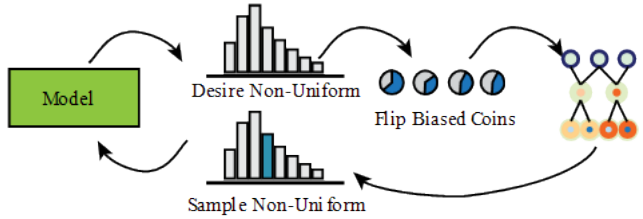
Step 2: Draw suitable model-specific RNs from hardware

COINFLIPS

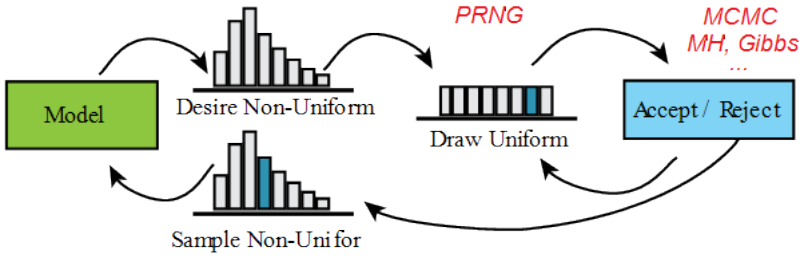
Evaluate opportunity of a probabilistic computing paradigm

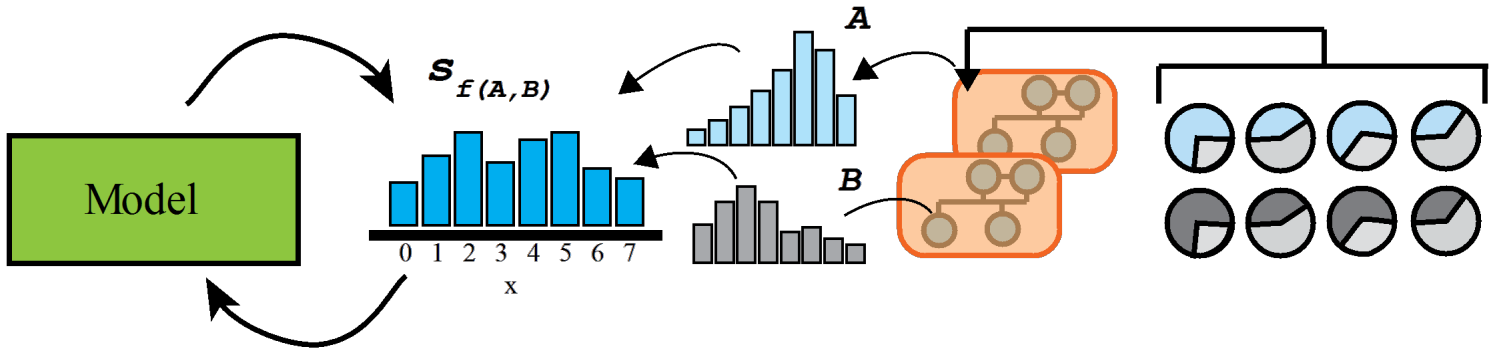
Future?





Today

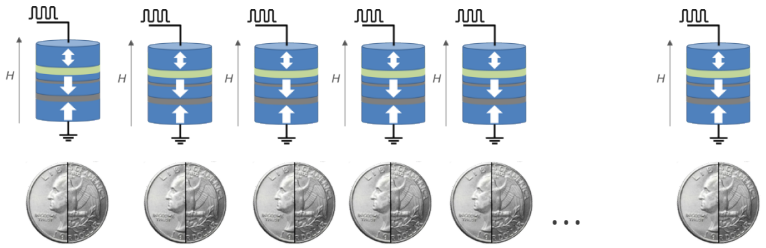




COINFLIPS

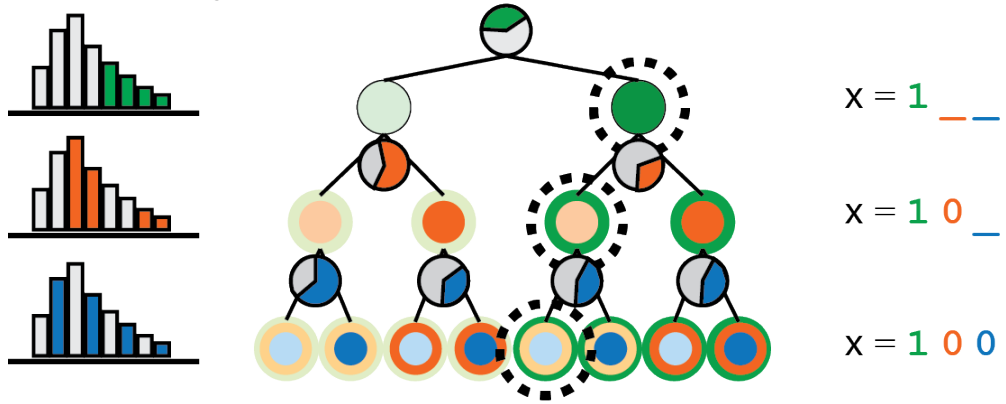
Step 3: Integrate hardware-enabled random sampling into computation

Mapping Coinflips to Arbitrary Distributions



Many devices flipping at one time

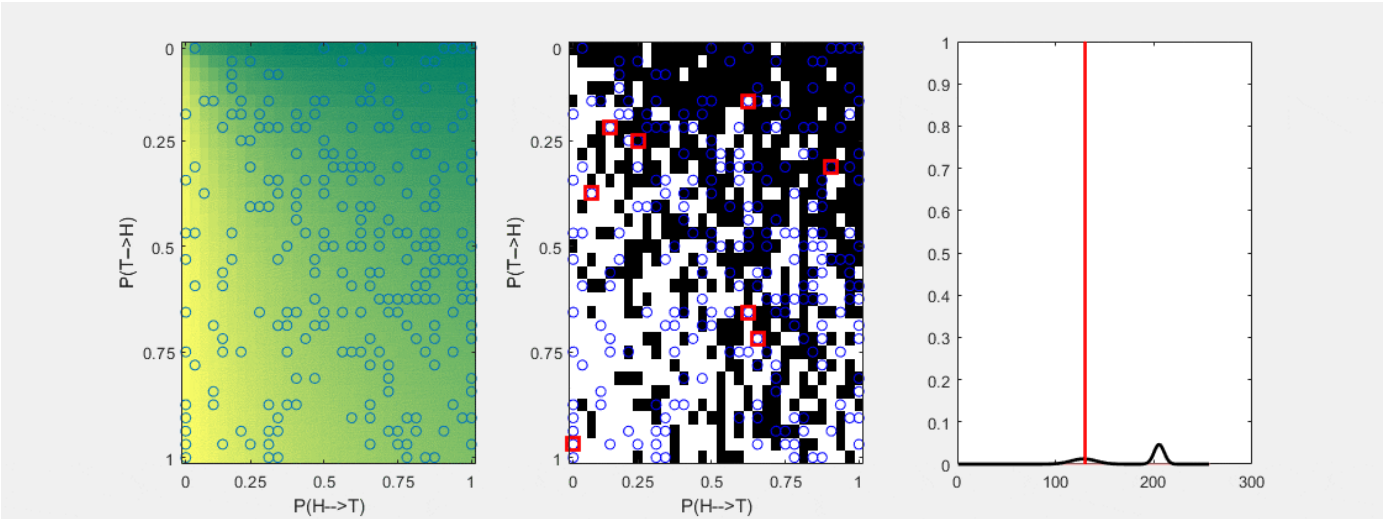
Naively, we can expand a binary tree with probabilities to describe any distribution



x = 1 _ _

x = 1 0 _

x = 1 0 0

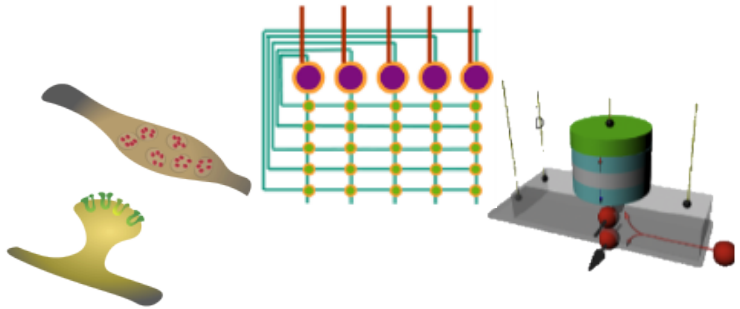


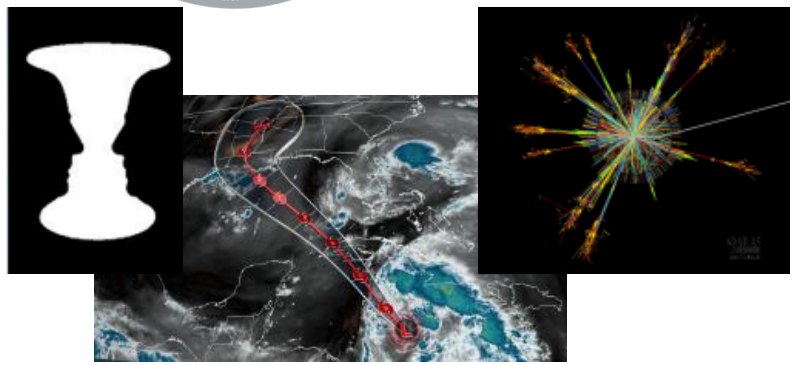
Simulation…

Probabilistic Neural Theory and Algorithms

Summary: Probabilistic computing is perhaps an ideal target for exploring potential for microelectronics co-design

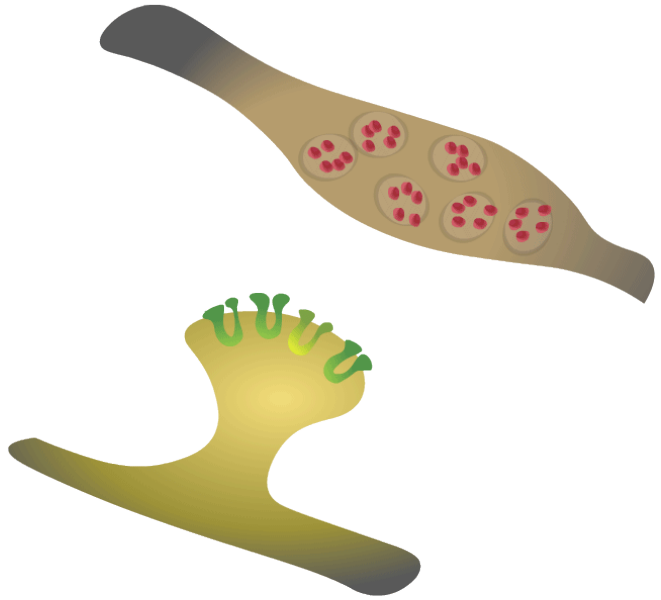
- All aspects of microelectronics (from materials to applications) have something to contribute
  - *Can show benefits from innovation at all scales*
- Stochastic devices
  + neuromorphic parallelism
  = broad application impact
  - *Both Mod-Sim and AI stand to benefit*
- Opportunity to consider important aspects of computing up front
  - *Address issues such as I/O, programmability, and theory from the onset, as opposed to after-the-fact*

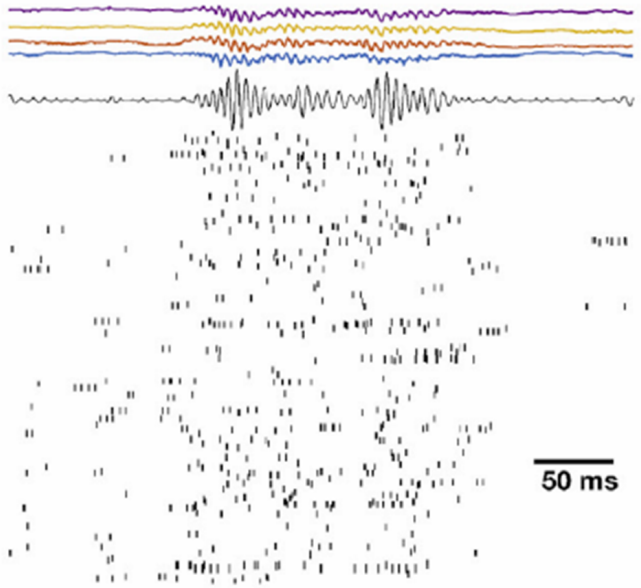


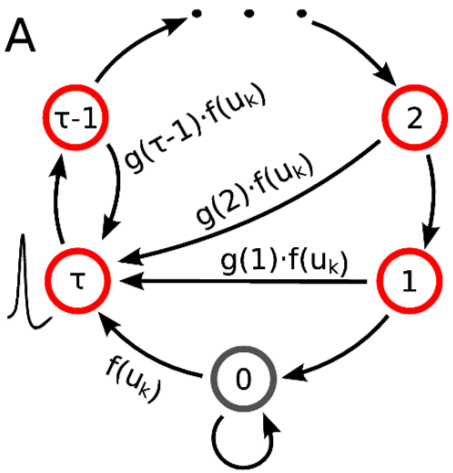


Backup: Neuroscience and stochastic computation

There is a long history of viewing neuroscience through a stochastic computation perspective. Most of this history is independent of envisioned computing applications







"Independent sources of quantal variability at single glutamatergic synapses" Franks KM, Stevens CF, Sejnowski TJ J. Neurosci. 2003

"Hippocampal Reactivation of Random Trajectories Resembling Brownian Diffusion" Stella F et al., Neuron. 2017

"Neural Dynamics as Sampling: A model of stochastic computation ..." Buesing L et al., PLOS Computational Biology. 2011

Has the tremendous success of deterministic computing left probabilistic applications behind?

Stochasticity reveals contrast in computing approaches

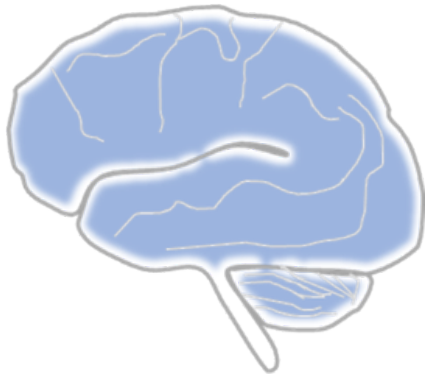
- Modern microelectronics spends tremendous resources in enforcing determinism
- The brain embraces and controls stochasticity across spatial and time scales

Developing probabilistic computing to address probabilistic applications

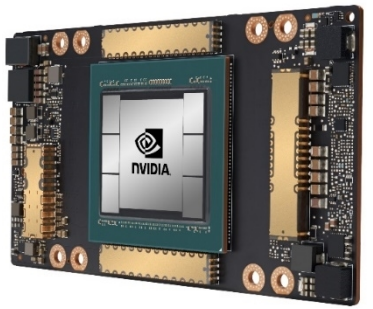
- ***COINFLIPS*** is combining stochastic devices with neuromorphic architectures
- Co-design is proving invaluable in developing this novel paradigm for microelectronics

Which approach is best to interpret an ambiguous input?

~20 W
~10^{15} synaptic events / second
Fully stochastic



~400 W
~10^{13}-10^{14} FLOPS
Fully deterministic



Thanks